

-
- CAS (v1v2v3)
 - 1
 - 2Cas
 - 2.1cas-client-coreJava
 - 2.1.1Spring MVC
 - 2.2.2Spring Boot
 - 2.2.3Spring Boot
 - 2.2shiro-casShiro
 - 2.3spring-security-casSpring Security
 - 3
 - 3.1
- OpenID
 - 1
 - 2OpenID Connect
 - 3OpenID
- OpenID Connect
- OAuth2.0
 - 1
 - 2Authorization Code
 - code
 - access_token
 - 3 Implicit
 - access_token
- Spring MVC

4Password Credentials

access_token

5Client Credentials

access_token

6OAuth2.0

6.1Apache Oltu

6.1.1Spring MVC

6.2.2Spring Boot

Spring Oauth2 Client

6

6.1

WS Federation

SAML(v1v2)

1

1Cas

2OAuth2.0

Cas

Markdown

Markdown

Markdown

Markdown

Markdown Mermaid ()

Markdown

Markdown Mermaid ()

Markdown

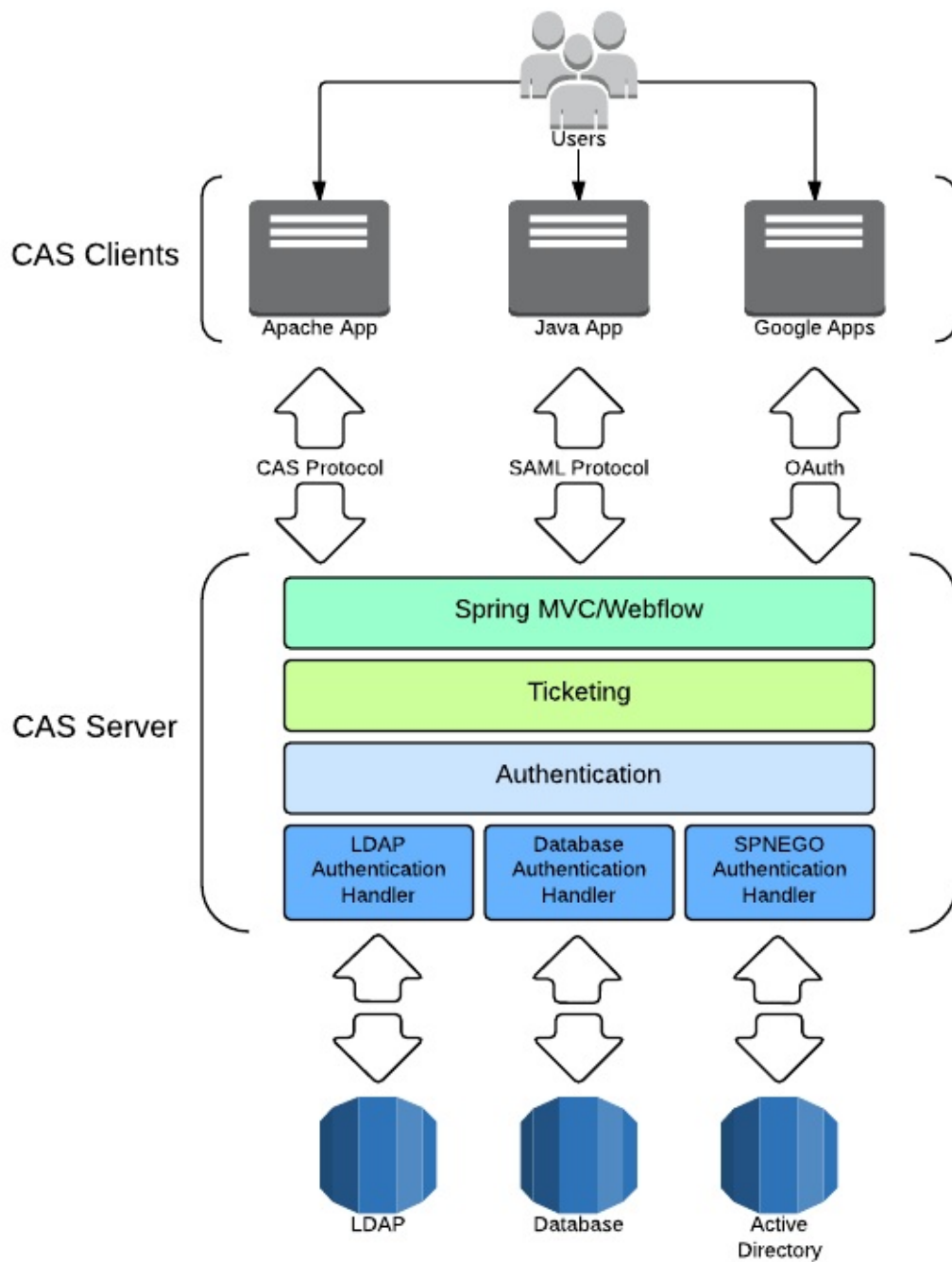
Markdown

Apereo CAS Web
CASJava

2

-
-
- PC

3Apereo CAS



CASCAS

3.1CAS

CASSpringJava ServletCASCASGTSSOTGTSTCASSTCAS

CAS_SERVER_URL CAS
CAS_SERVER_URL = <http://sso.hiwepy.com>

3.2CAS

“ CAS client” CASCASCASCASSAMLOAuthCASCAS

```
MY_SERVER_URL CAS
MY_SERVER_URL = http://client.hiwepy.com
```

Platforms:

- Apache httpd Server (mod_auth_cas module)
- Java (Java CAS Client)
- .NET (.NET CAS Client)
- PHP (phpCAS)
- Perl (PerlCAS)
- Python (pycas)
- Ruby (rubycas-client)

Applications:

- Canvas
- Atlassian Confluence
- Atlassian JIRA
- Drupal
- Liferay
- uPortal
- ...

ipURLhttp://
t.edu.cnCAS ClientOAuth http://t.edu.cn/main.dohttp://t.edu.cn/login.htm
http://t2.zju.edu.cn
appkeyappsecret

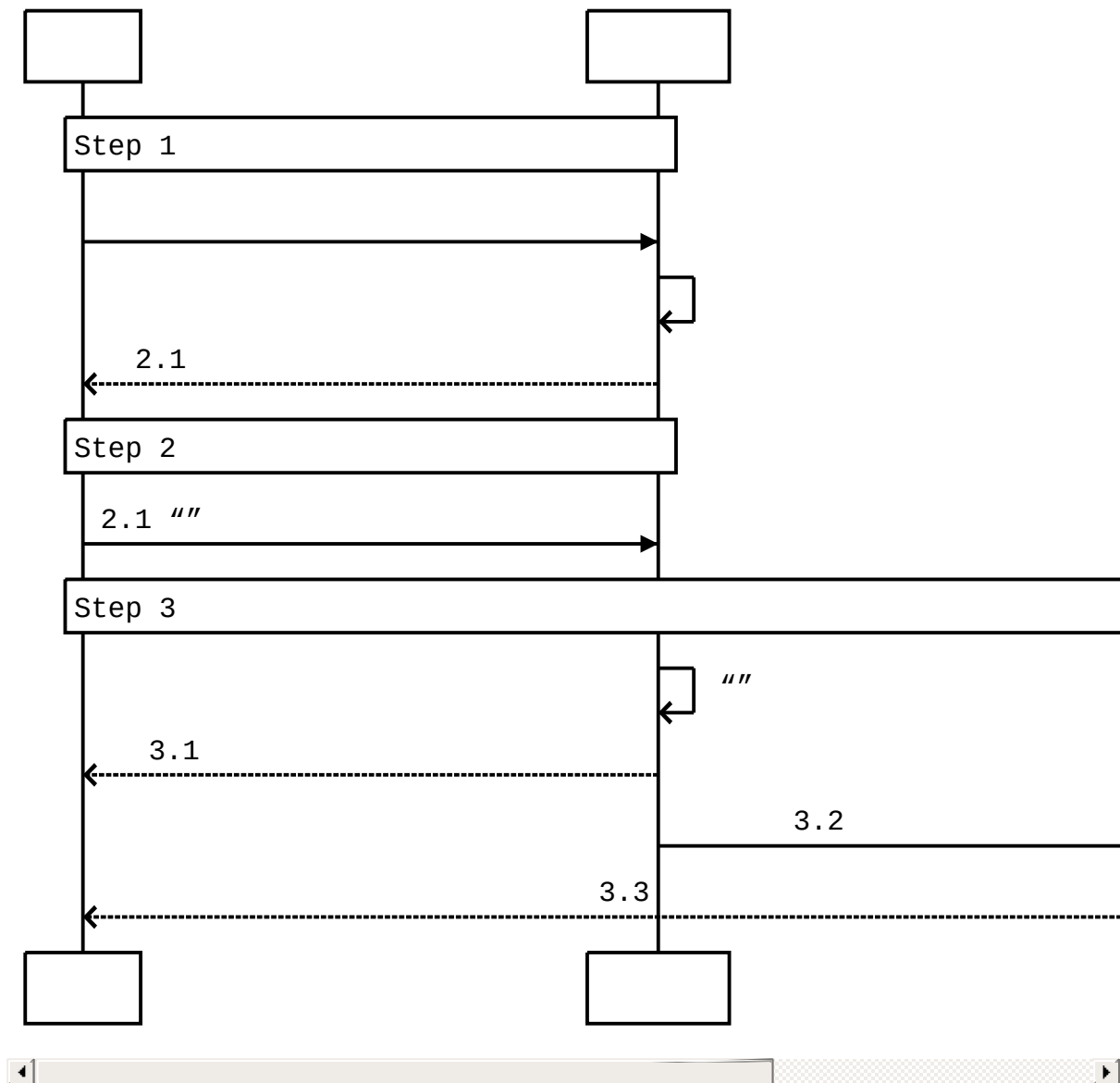
1

https://cas_server_url	https://xlfsso.hzgsedu.cn

/



-
-
-
-



1. CAS

CAS Central Authentication Service Web

TGTTicket Granting Ticket

TGT CAS TGT CAS TGT Cookie Cookie HTTP CAS
Cookie TGC key TGT

TGCTicket Granting Cookie

CAS Server TGT Session TGC Session SessionId Cookie CAS Server

STService Ticket

ST CAS service service service ST CAS ST CAS ST CAS
TGT ST ST serviceservice ST CAS

1.1 CAS

CAS
CAS ServerOAuth2+JWT /
CAS Client

1.2

- CAS Client
- Apache httpd Server (mod_auth_cas module)
- Java (Java CAS Client)
- .NET (.NET CAS Client)
- PHP (phpCAS)
- Perl (PerlCAS)
- Python (pycas)Ruby (rubycas-client)

2. OAuth2.0

OAuth Open Authentication

2.1 OAuth 2.0

- resource owner): appapp
- clientapp
- Authorization server:
- Resource server
- Authorization grant
- Redirect UriUri
- access token
- Scopeapp

2.2 OAuth2.0

-
-
-
-
-
-

2.3 OAuth2.0

2.3.1. Authorization Code

2.3.2. Implicit

JavaScript

2.3.3. Resource Owner Password Credentials

2.3.4. Client Credentials

CAS

Cas XML

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
  <cas:authenticationSuccess>
    <cas:user>mr100758</cas:user>
    <cas:attributes>
      <cas:infoId>1782646895357755392</cas:infoId>
      <cas:gender>1</cas:gender>
      <cas:orgList>[{"orgId":1781203561347432450,"orgType":,"node_school":,"orgCode":330108060002,"orgName":,"stageCode":211,"campusList":[{"orgId":1781203562983211009,"orgCode":33010806000201,"orgName":,"orgId":1783373046791684098,"orgCode":33010806000202,"orgName":,"orgId":1783754135544676353,"orgCode":33010806000203,"orgName":,"orgId":1785169957806817281,"orgCode":33010806000204,"orgName":,"identityList":[{"userId":1782646895655550978,"identityId":1,"identityName":,"infoId":1782646895357755392}],roleList":[{"orgId":1781203561347432450,"roleId":8,"roleCode":,"js_000008","roleName":,"identityId":1}]}</cas:orgList>
      <cas:childList>[{"accountId":1781224602786103298,"account":,"bts06123233","userId":1781224602257620994,"id":1781224602144374784,"xm":,"xbm":,"xsdqz":,"xxdm":,"xxmc":,"xxb":,"xqdm":,"xqmc":,"nj":,"njmc":,"bh":,"202302110101","bjmc":,"01"}]</cas:childList>
      <cas:type>password</cas:type>
      <cas:xsId>1781224602144374784</cas:xsId>
      <cas:orgId>1781203561347432450</cas:orgId>
      <cas:loginTime>1715117492</cas:loginTime>
      <cas:nickname> </cas:nickname>
      <cas:longTermAuthenticationRequestTokenUsed>>false</cas:longTermAuthenticationRequestTokenUsed>
      <cas:email></cas:email>
      <cas:clientIpAddress>115.193.187.209</cas:clientIpAddress>
      <cas:isFromNewLogin>>false</cas:isFromNewLogin>
```

```

        <cas:authenticationDate>2024-05-07T21:31:44.957+08:00[Asia/Shanghai]</cas:authenticationDate>
        <cas:avatarUrl></cas:avatarUrl>
        <cas:successfulAuthenticationHandlers>LoginAuthenticationHandler</cas:successfulAuthenticationHandlers>
        <cas:xsUserId>1781224602257620994</cas:xsUserId>
        <cas:xqOrgId>1781203562983211009</cas:xqOrgId>
        <cas:userId>178264689565550978</cas:userId>
        <cas:credentialType>LoginCredential</cas:credentialType>
        <cas:accountId>1782646897865949186</cas:accountId>
        <cas:authenticationMethod>LoginAuthenticationHandler</cas:authenticationMethod>
        <cas:identityId>3</cas:identityId>
        <cas:account>mr100758</cas:account>
        </cas:attributes>
    </cas:authenticationSuccess>
</cas:serviceResponse>

```

Assertion JSON

```

{
  "attributes": {
  },
  "authenticationDate": "1715088719697",
  "principal": {
    "attributes": {
      "clientIpAddress": "115.193.187.209",
      "infoId": "1782646895357755392",
      "isFromNewLogin": "false",
      "gender": "1",
      "authenticationDate": "2024-05-07T21:31:44.957+08:00[Asia/Shanghai]",
      "avatarUrl": "",
      "successfulAuthenticationHandlers": "LoginAuthenticationHandler",
      "xsUserId": "1781224602257620994",
      "xqOrgId": "1781203562983211009",
      "orgList": "[{\"orgId\":\"1781203561347432450\",\"orgType\":\"node_school\",\"orgCode\":\"330108060002\",\"orgName\":\"\",\"stageCode\":\"211\",\"campusList\": [{\"orgId\":\"1781203562983211009\",\"orgCode\":\"33010806000201\",\"orgName\":\"\"}, {\"orgId\":\"1783373046791684098\",\"orgCode\":\"33010806000202\",\"orgName\":\"2\"}, {\"orgId\":\"1783754135544676353\",\"orgCode\":\"33010806000203\",\"orgName\":\"3\"}, {\"orgId\":\"1785169957806817281\",\"orgCode\":\"33010806000204\",\"orgName\":\"4\"}], \"identityList\": [{\"userId\":\"178264689565550978\",\"identityId\":\"1\",\"identityName\":\"\",\"infoId\":\"1782646895357755392\"}], \"roleList\": [{\"orgId\":\"1781203561347432450\",\"roleId\":\"8\",\"roleCode\":\"js_000008\",\"roleName\"

```

```

":\\""}],\\"identityId\\":1}]" ,
    "childList": "[\\"accountId\\":1781224602786103298,\\"account\\":\\"bts06123233\\",\\"userId\\":1781224602257620994,\\"id\\":1781224602144374784,\\"xm\\":\\"\\",\\"xbm\\":\\"1\\",\\"xsdqztm\\":\\"1\\",\\"xxdm\\":\\"330108060002\\",\\"xxmc\\":\\"\\",\\"xxbxmlm\\":\\"211\\",\\"xqdm\\":\\"33010806000201\\",\\"xqmc\\":\\"\\",\\"nj\\":\\"11\\",\\"njmc\\":\\"\\",\\"bh\\":\\"202302110101\\",\\"bjmc\\":\\"01\\"}]" ,
    "type": "password",
    "userId": "1782646895655550978",
    "xsId": "1781224602144374784",
    "orgId": "1781203561347432450",
    "credentialType": "LoginCredential",
    "accountId": "1782646897865949186",
    "loginTime": "1715117492",
    "authenticationMethod": "LoginAuthenticationHandler",
    "identityId": "3",
    "nickname": "" ,
    "longTermAuthenticationRequestTokenUsed": "false",
    "email": "",
    "account": "mr100758"
  },
  "name": "mr100758"
},
"valid": "true",
"validFromDate": "1715088719697"
}

```

loginType	string	passworddingtalk-scancode
loginTime	integer(int64)	
account	string	
accountId	integer(int64)	ID
accountStatus	integer	0:
userId	integer(int64)	ID
userName	string	
userCode	string	(/)

userStatus	integer	0:
nickname	string	
avatarUrl	string	
phone	string	
email	string	
gender	integer	12
clientIpAddress	string	IP
childAccountList	json	List<UserChildProfileV0> JSON
orgList	json	List<UserOrgV0> JSON
orgId	integer(int64)	ID
xqOrgId	integer(int64)	ID
identityId	integer(int64)	(1:2:3:4:5:6:7:)
roleId	integer(int64)	ID
xsId	integer(int64)	ID
xsUserId	integer(int64)	ID

Schema :

UserChildProfileVO

			schema
accountId	ID	integer(int64)	
account		string	
userId	ID	integer(int64)	
id	ID	integer(int64)	1781224602144374784
xm		string	
xh		string	
zp		string	

xbm	1:,2:	string	
xsdqztn	1:,2:,3:,4:,5:,6:,7:,9:	string	
xxdm		string	330108060002
xxmc		string	
xxbxxm	111:,211:,311:,312:,342:,365:,513:,999:,998:,997:,341:	string	
xqdm		string	33010806000201
xqmc		string	
nj		string	
njmc		string	
bh		string	
bjmc		string	

UserOrgVO

			schema
orgId	IDID	integer(int64)	
orgType	//	string	
orgCode	//	string	
orgName		string	
stageCode	111211311 312342365 368513999	integer	
campusList	/	json	List<UserCampusVO> JSON
identityList	/	json	List<UserIdentityVO> JSON
			List<UserRoleVO>

			JSON
--	--	--	------

UserCampusVO

			schema
orgId	IDID	integer(int64)	
orgCode		string	
orgName		string	

UserIdentityVO

			schema
identityId		integer(int64)	
identityName		string	

UserRoleVO

			schema
roleId	Id	integer(int64)	
roleCode		string	
roleName		string	

-

-

1

•

•

•

•

+ +

•

•

•

/

/

123456

+ +

2

+ +

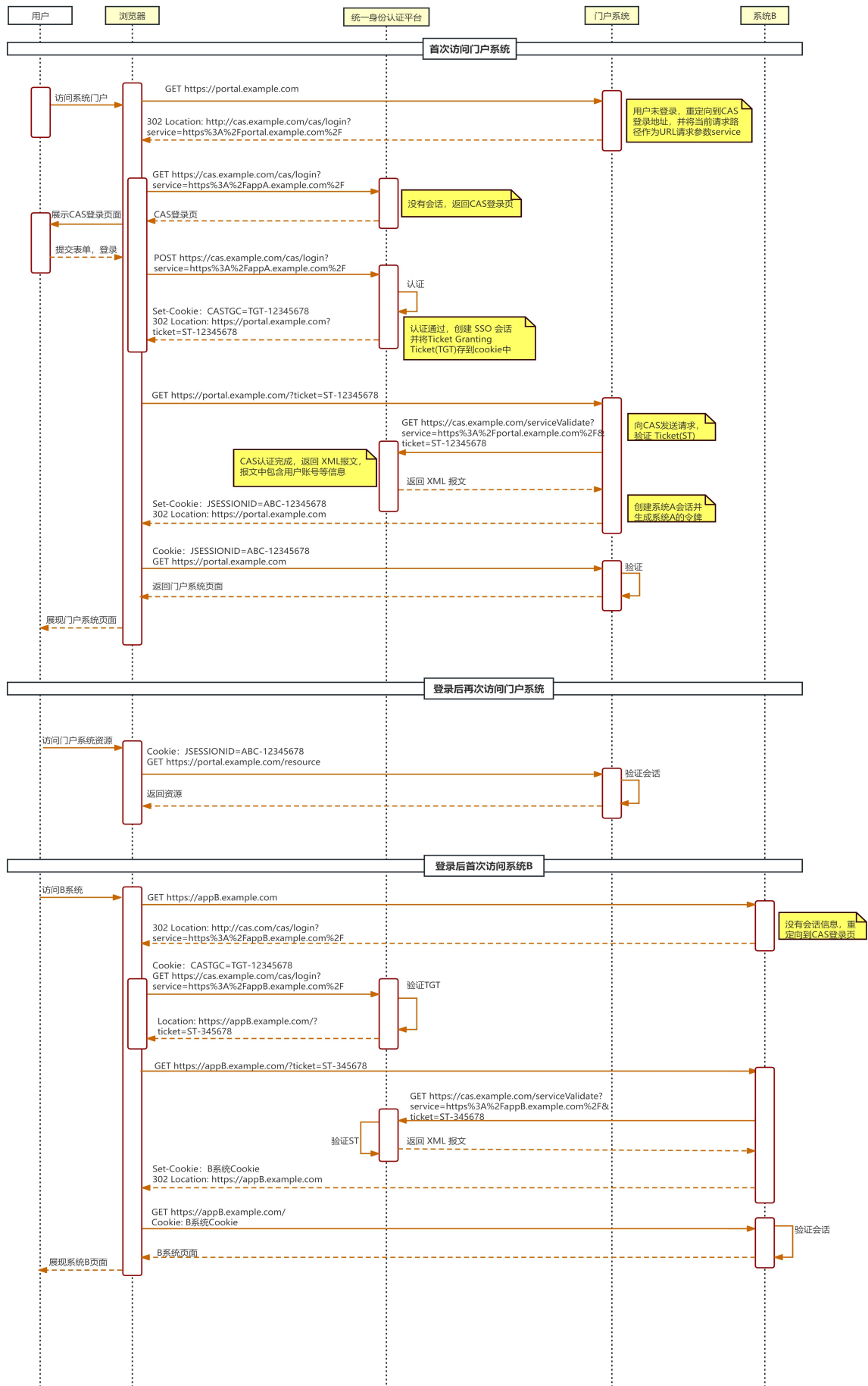
+ +

• Cas

•

•

3



CAS

1 1

- 1 Redis orgIdidentityIdroleId
- 2 Redis subAccountsubAccountIdsubUserIdsubOrgId
subSchoolCodesubGradeCodesubClassCode

2 CAS 21

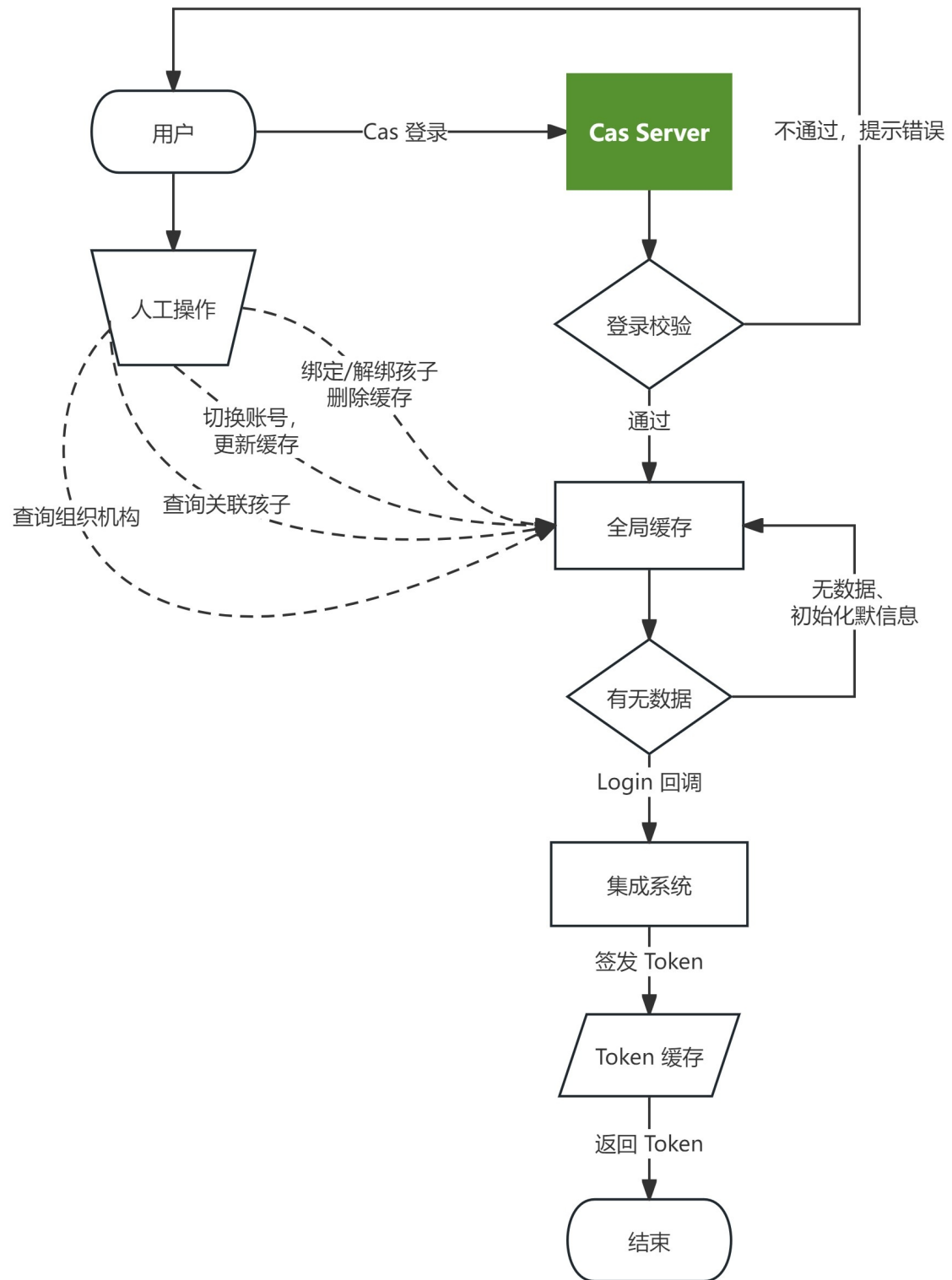
- 1 [CAS] orgIdidentityIdroleId subAccountsubAccountId
subUserIdsubOrgIdsubSchoolCodesubGradeCodesubClassCode
- 2 [CAS] /p3/validate Redis
- 3 [CAS] OAuth2.0 /oauth2.0/profile

3 1

-

4 1

- Auth Cas Cas
- Auth Cas Cas

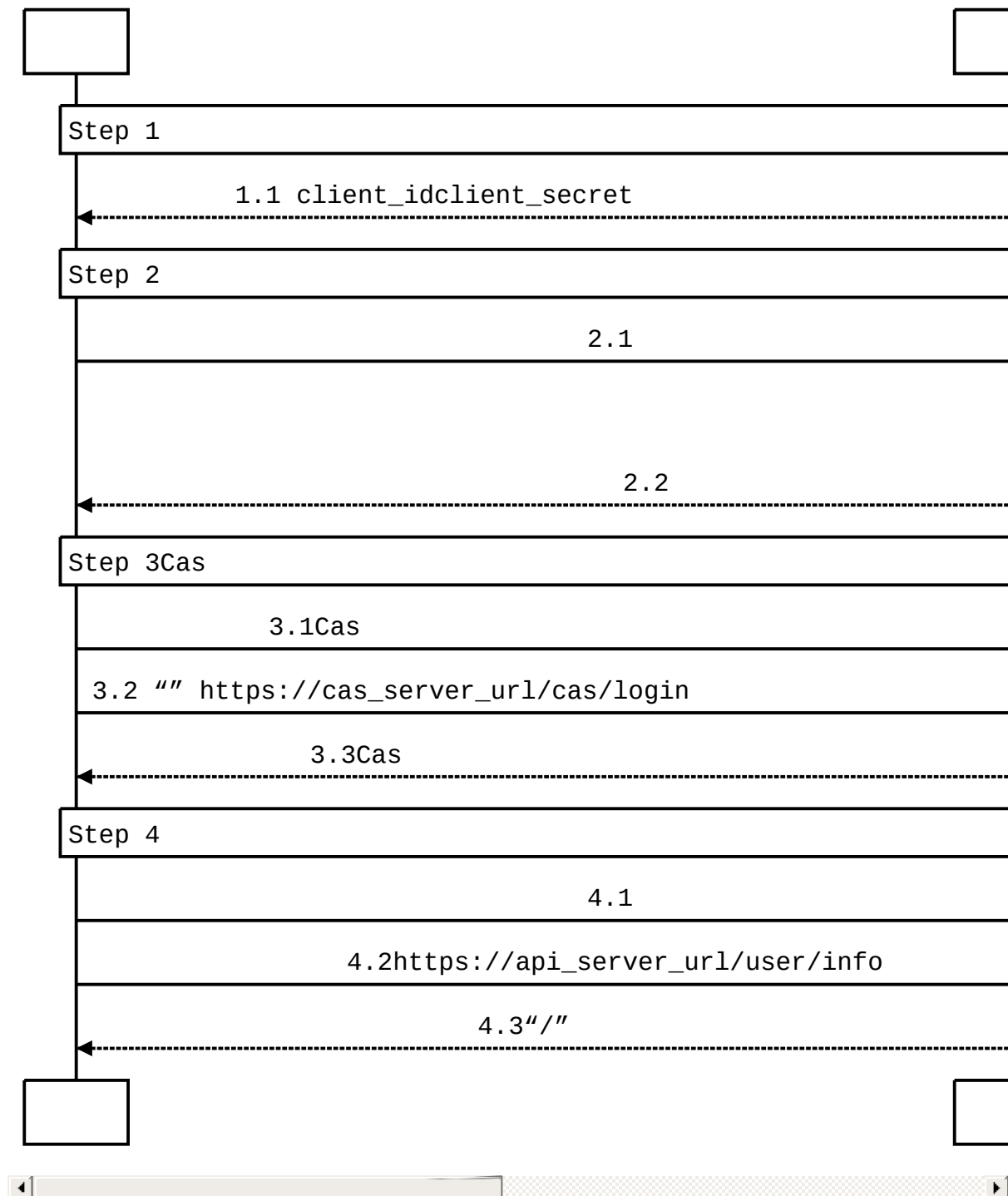


CAS (v1v2v3)

1

Cas

CAS - B/S Cookie //



2Cas

Cas

Cas

- Apache httpd Server ([mod_auth_cas module](#))
- Java ([Java CAS Client](#))
- .NET ([.NET CAS Client](#))
- PHP ([phpCAS](#))
- Perl ([PerlCAS](#))
- Python ([pycas](#))
- Ruby ([rubycas-client](#))
- Shiro ([shiro-cas](#))
- Spring Security ([spring-security-cas](#))

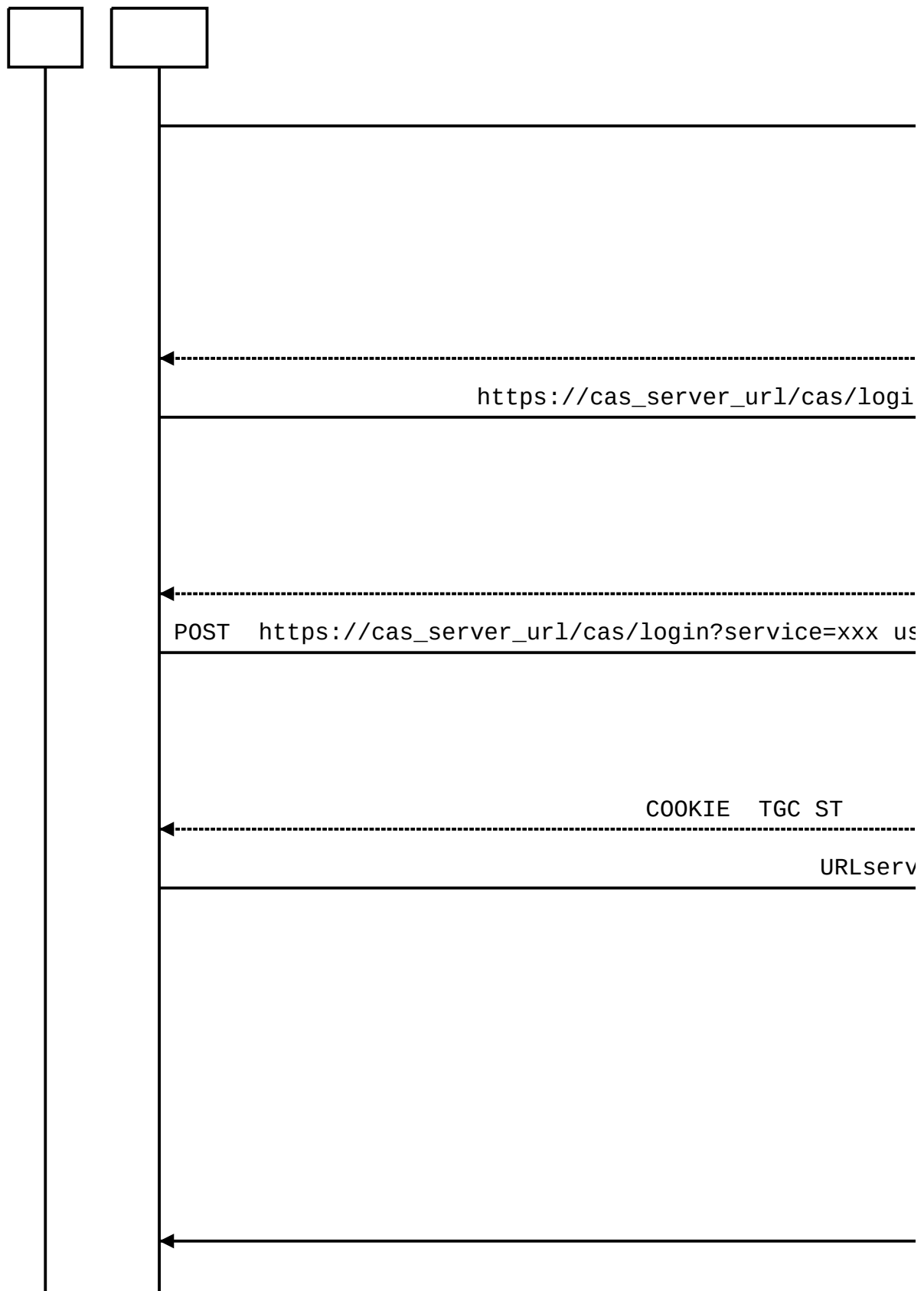
cas-client-core	Java Web	https://github.com/apereo/java-cas-client	
phpCAS	PHP Web	https://github.com/apereo/phpCAS	
.NET CAS Client	.NET Web	https://github.com/apereo/dotnet-cas-client	
shiro-cas	shiroJava Web	https://github.com/hiwepy/shiro-cas-spring-boot-starter	
spring-security-cas	spring-securityJava Web		

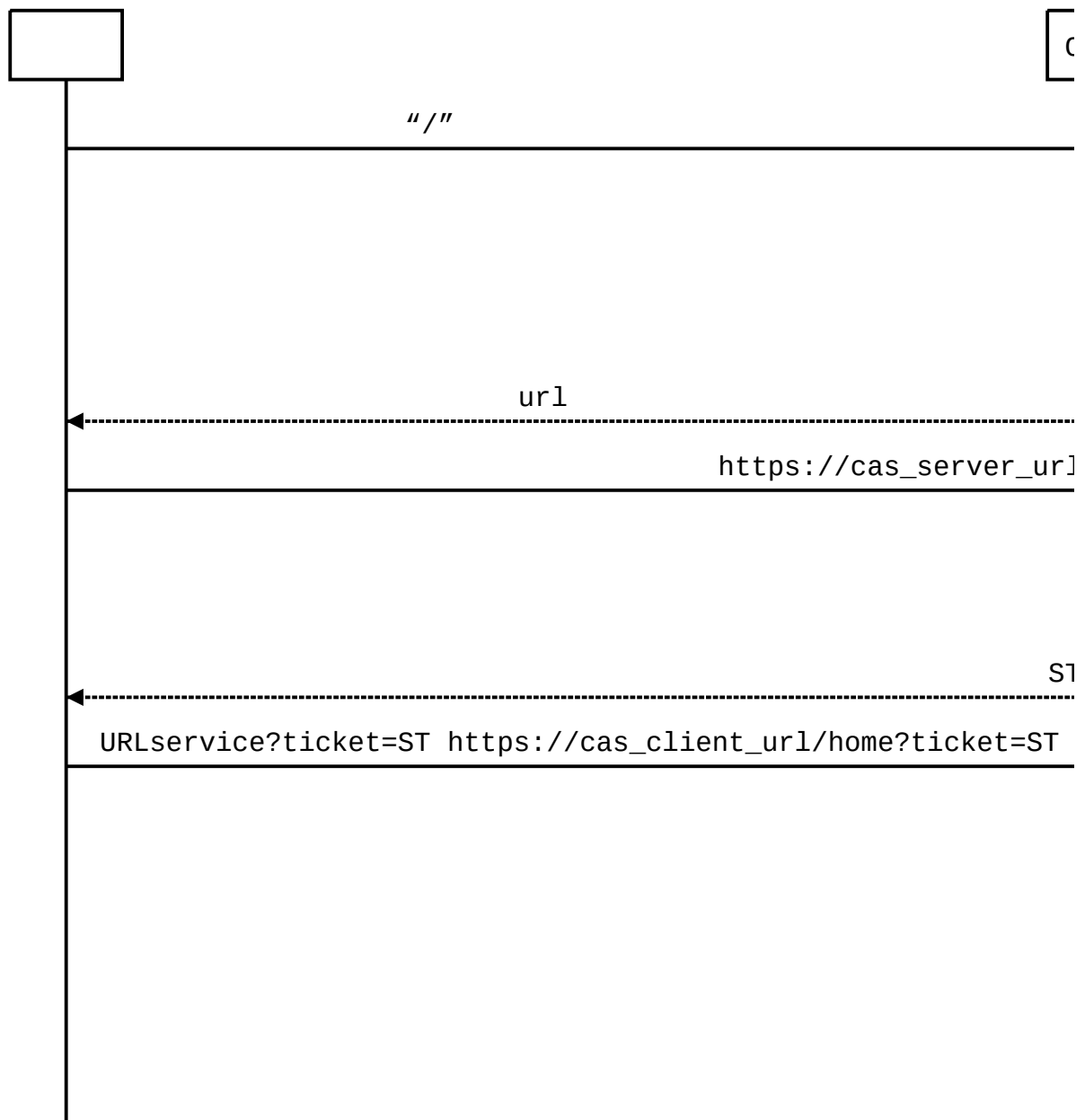
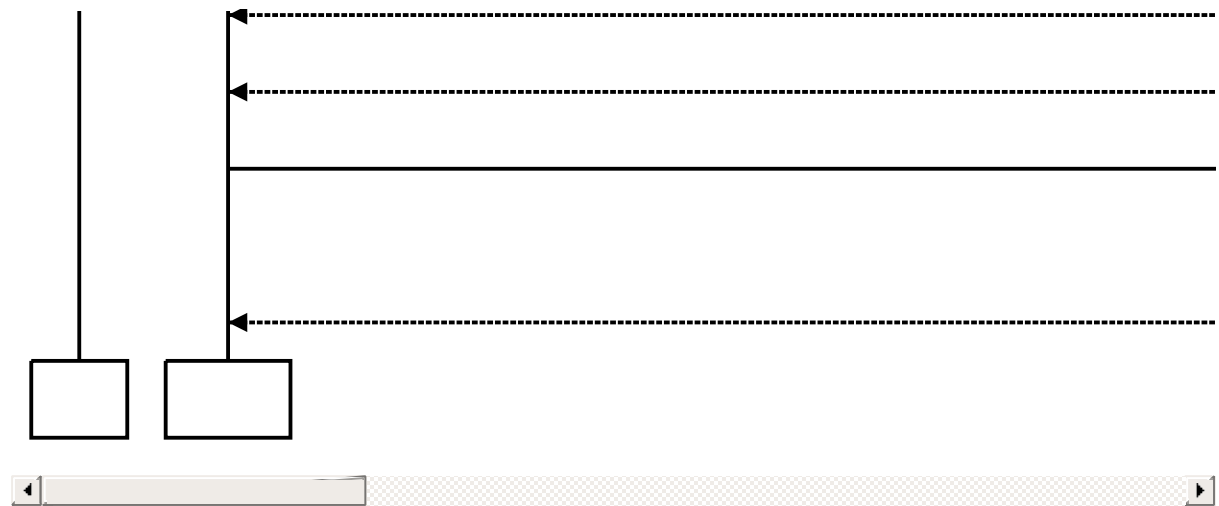
Cas

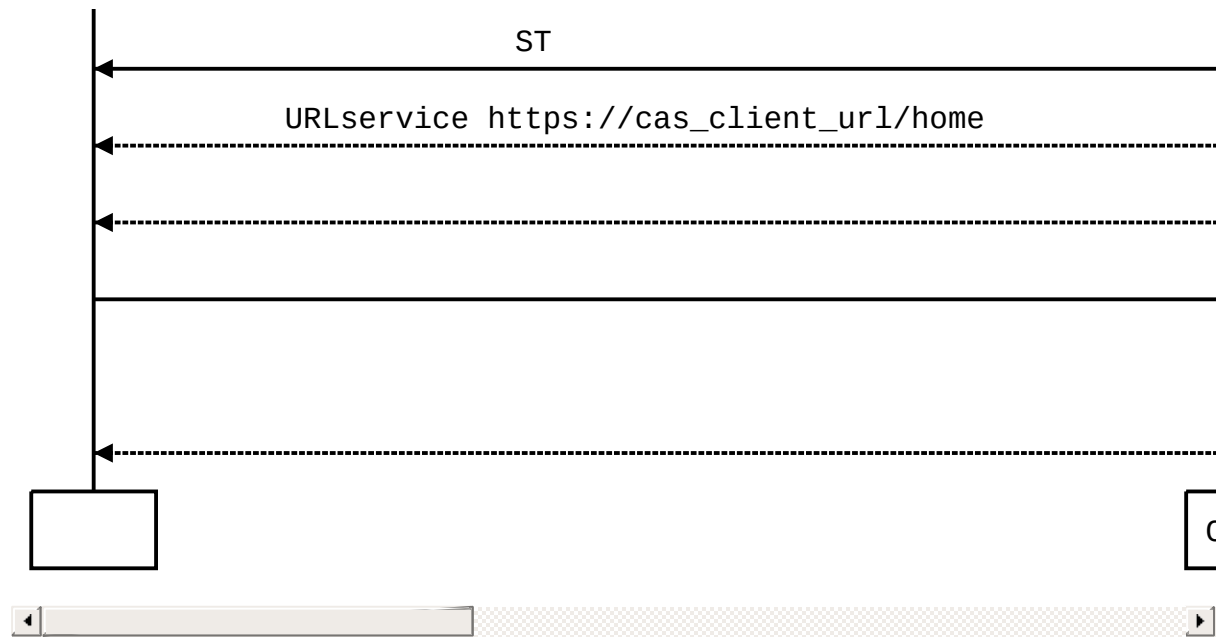
CAS Ticket Ticket

STService Ticket
 ST CAS service CAS url
 TGTicket Granting Ticket
 TGT CAS TGT
 TGTicket Granted Cookie) TGT

LTLogin Ticket







2.1cas-client-coreJava

cas-client-core Cas SDK

Cas

- Spring MVC Web Cas
- Spring Boot Cas
- Spring Boot Cas

2.1.1Spring MVC

Spring MVC Web Cas

cas-client-core Cas SDK

1

cas-client-core

Maven

```
<!-- https://mvnrepository.com/artifact/org.jasig.cas.client/cas-client-core -->
<dependency>
  <groupId>org.jasig.cas.client</groupId>
  <artifactId>cas-client-core</artifactId>
  <version>3.6.4</version>
</dependency>
```

Gradle

```
// https://mvnrepository.com/artifact/org.jasig.cas.client/cas-client-core
implementation group: 'org.jasig.cas.client', name: 'cas-client-core', version: '3.6.4'
```

2

webapp web.xml,

```
<listener>
  <listener-class>
    org.jasig.cas.client.session.SingleSignOutHttpSessionListener
  </listener-class>
</listener>
<!-- --->
<filter>
  <filter-name>CAS Single Sign Out Filter</filter-name>
  <filter-class>org.jasig.cas.client.session.SingleSignOutFilter</filter-class>
</filter>
```

```

<filter-mapping>
  <filter-name>CAS Single Sign Out Filter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
<!-->
<filter>
  <filter-name>CAS Authentication Filter</filter-name>
  <filter-class>org.jasig.cas.client.authentication.AuthenticationFilter
</filter-class>
  <!-->
  <init-param>
    <param-name>casServerLoginUrl</param-name>
    <!-->
    <param-value>https://cas_server_url/login</param-value>
  </init-param>
  <!--ip-->
  <init-param>
    <param-name>serverName</param-name>
    <!--,http://ip:-->
    <param-value>http://ip:  </param-value>
  </init-param>
  <init-param>
    <param-name>acceptAnyProxy</param-name>
    <param-value>true</param-value>
  </init-param>
</filter>

<filter>
  <filter-name>CAS Validation Filter</filter-name>
  <filter-class>org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter</filter-class>
  <!-->
  <init-param>
    <param-name>casServerUrlPrefix</param-name>
    <param-value>https://cas_server_url</param-value>
  </init-param>
  <!--ip-->
  <init-param>
    <param-name>serverName</param-name>
    <!--,http://ip:-->
    <param-value>http://ip:  </param-value>
  </init-param>
</filter>

<filter>
  <filter-name>CAS HttpServletRequest Wrapper Filter</filter-name>
  <filter-class>org.jasig.cas.client.util.HttpServletRequestWrapperFilter</filter-class>

```

```

</filter>

<filter>
  <filter-name>CAS Assertion Thread Local Filter</filter-name>
  <filter-class>org.jasig.cas.client.util.AssertionThreadLocalFilter</fi
lter-class>
</filter>

<filter-mapping>
  <filter-name>CAS Validation Filter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<filter-mapping>
  <filter-name>CAS Authentication Filter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<filter-mapping>
  <filter-name>CAS HttpServletRequest Wrapper Filter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<filter-mapping>
  <filter-name>CAS Assertion Thread Local Filter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<session-config>
  <tracking-mode>COOKIE</tracking-mode>
</session-config>
<!-->

```


2.2.2Spring Boot

Spring Boot Cas

cas-client-support-springboot Cas SDK Spring Boot

1

cas-client-support-springboot

Maven

```
<!-- https://mvnrepository.com/artifact/org.jasig.cas.client/cas-client-support-springboot -->
<dependency>
  <groupId>org.jasig.cas.client</groupId>
  <artifactId>cas-client-support-springboot</artifactId>
  <version>3.6.4</version>
</dependency>
```

Gradle

```
// https://mvnrepository.com/artifact/org.jasig.cas.client/cas-client-support-springboot
implementation group: 'org.jasig.cas.client', name: 'cas-client-support-springboot', version: '3.6.4'
```

2

2.1 AntUrlPatternMatcherStrategy Ant

```
import org.jasig.cas.client.authentication.UrlPatternMatcherStrategy;
import org.springframework.util.AntPathMatcher;
import org.springframework.util.StringUtils;

public class AntUrlPatternMatcherStrategy implements UrlPatternMatcherStrategy {

    /**
     * Any number of these characters are considered delimiters between
     * multiple
     * context config paths in a single String value.
     */
}
```

```

public static String CONFIG_LOCATION_DELIMITERS = ",; \\t\\n";
private AntPathMatcher matcher = new AntPathMatcher();
private String[] patterns;

@Override
public boolean matches(String url) {
    for (String pattern : patterns) {
        if (matcher.match(pattern, url)) {
            return true;
        }
    }
    return false;
}

@Override
public void setPattern(String pattern) {
    this.patterns = StringUtils.tokenizeToStringArray(pattern, CONFIG_LOCATION_DELIMITERS);
}
}

```

2.2SSL HostnameVerifier

```

public class TrustAllHostnameVerifier implements HostnameVerifier {

    public final static TrustAllHostnameVerifier DEFAULT = new TrustAllHostnameVerifier();

    @Override
    public boolean verify(String hostname, SSLSession session) {
        return true;
    }
}

```

2.3CasCustomCasClientConfigurer

```

import org.jasig.cas.client.boot.configuration.CasClientConfigurer;
import org.jasig.cas.client.configuration.ConfigurationKeys;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.boot.web.servlet.FilterRegistrationBean;
import org.springframework.stereotype.Component;

import java.util.Map;

/**
 * cas-client-support-springboot CAS

```

```

    * {                                     @link CasClientConfigurer}
    */
@Component
public class CustomCasClientConfigurer implements CasClientConfigurer {

    /**
     * CAS-protected client Ignore Pattern Path, E.g: /api/*,/auth/*
     */
    @Value("${cas-ignore-pattern-path}")
    private String ignorePatternPath;

    /**
     * /cas/logout
     */
    @Override
    @SuppressWarnings({ "rawtypes", "unchecked" })
    public void configureAuthenticationFilter(final FilterRegistrationBean authenticationFilter) {
        Map initParameters = authenticationFilter.getInitParameters();
        initParameters.put(ConfigurationKeys.IGNORE_URL_PATTERN_TYPE.getName(), AntUrlPatternMatcherStrategy.class.getName());
        initParameters.put(ConfigurationKeys.IGNORE_PATTERN.getName(), ignorePatternPath);
    }

    @Override
    public void configureValidationFilter(FilterRegistrationBean validationFilter) {
        Map initParameters = validationFilter.getInitParameters();
        initParameters.put(ConfigurationKeys.HOSTNAME_VERIFIER.getName(), TrustAllHostnameVerifier.class.getName());
        initParameters.put(ConfigurationKeys.IGNORE_PATTERN.getName(), ignorePatternPath);
    }
}

```

Cas

Cas Cas

1 CasAuthController

/cas/index /cas/login /cas/logout /cas/logoutPage 4 Cas

```

import io.swagger.annotations.Api;
import io.swagger.annotations.ApiOperation;
import lombok.extern.slf4j.Slf4j;

```

```

import org.apache.commons.collections4.MapUtils;
import org.jasig.cas.client.Protocol;
import org.jasig.cas.client.boot.configuration.CasClientConfigurationProperties;
import org.jasig.cas.client.util.AbstractCasFilter;
import org.jasig.cas.client.util.CommonUtils;
import org.jasig.cas.client.validation.Assertion;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.util.*;

@Api(tags = "Cas ")
@RestController
@RequestMapping("cas")
@Slf4j
public class CasAuthController {

    @Autowired
    private CasClientConfigurationProperties casProperties;

    /**
     *
     */
    @GetMapping("/index")
    @ResponseBody
    public Map<String, Object> index(HttpServletRequest request, HttpServletResponse response) throws IOException {
        // 1CAS
        HttpSession session = request.getSession(false);
        Assertion assertion = (Assertion) (session == null ? request
            .getAttribute(AbstractCasFilter.CONST_CAS_ASSERTION) : session
            .getAttribute(AbstractCasFilter.CONST_CAS_ASSERTION));
        // 2
        if (Objects.isNull(assertion)) {
            // 2.1
            response.sendRedirect(casProperties.getClientHostUrl() + "/cas/login");
            return Collections.emptyMap();
        }
        // 3
        Map<String, Object> userInfo = CasUtil.getUserAttributes();
        Map<String, Object> result = new HashMap<>(3);

```

```

        result.put("code", 200);
        result.put("msg", MapUtils.getString(userInfo, "nickname", "xx")
+ ", "      );
        result.put("data", "<a href=\"" + casProperties.getClientHostUrl
() + "/cas/logout\"></a>"      );
        return result;
    }

    /**
     * CAS Server
     */
    @GetMapping("login")
    @ApiOperation("Cas ")
    public void login(HttpServletRequest request, HttpServletResponse re
sponse) throws IOException {
        // 1CAS
        HttpSession session = request.getSession(false);
        Assertion assertion = (Assertion) (session == null ? request
            .getAttribute(AbstractCasFilter.CONST_CAS_ASSERTION) : s
ession
            .getAttribute(AbstractCasFilter.CONST_CAS_ASSERTION));
        // 2CAS Server
        if (Objects.isNull(assertion)) {
            // 2.1
            String casSingleLoginUrl = CommonUtils.constructRedirectUrl(
casProperties.getServerLoginUrl(),
                Protocol.CAS2.getServiceParameterName(),
                casProperties.getClientHostUrl() + "/cas/login",
                Boolean.FALSE, Optional.ofNullable(casProperties.get
Gateway()).orElse(Boolean.FALSE));
            response.sendRedirect(casSingleLoginUrl);
            return;
        }
        // 3
        final String urlToRedirectTo = casProperties.getClientHostUrl()
+ "/cas/index";
        if (log.isDebugEnabled()) {
            log.debug("redirecting to \"" + urlToRedirectTo + "\"");
        }
        response.sendRedirect(urlToRedirectTo);
    }

    /**
     *
     */
    @GetMapping("/logout")
    @ApiOperation("cas ")
    public void logout(HttpServletRequest request, HttpServletResponse r

```

```

response, @RequestParam(required = false) String url) throws IOException
{
    // 1
    HttpSession session = request.getSession(false);
    if (Objects.nonNull(session)) {
        // 1.1
        session.invalidate();
    }
    // 2CAS ServerURLservice
    String casSingleLoginUrl = CommonUtils.constructRedirectUrl(casP
    roperties.getServerUrlPrefix() + "/logout",
        Protocol.CAS2.getServiceParameterName(),
        casProperties.getClientHostUrl() + "/cas/logoutPage",
        Boolean.FALSE, Optional.ofNullable(casProperties.getGate
    way()).orElse(Boolean.FALSE));
    // 3CAS Server
    response.sendRedirect(casSingleLoginUrl);
}

/**
 *
 */
@GetMapping("/logoutPage")
@ResponseBody
public Map<String, Object> logoutPage(HttpServletResponse response)
{
    // The URL to the CAS Server Single login.
    String casSingleLoginUrl = CommonUtils.constructRedirectUrl(casP
    roperties.getServerLoginUrl(),
        Protocol.CAS2.getServiceParameterName(),
        casProperties.getClientHostUrl() + "/cas/login",
        Boolean.FALSE, Optional.ofNullable(casProperties.getGate
    way()).orElse(Boolean.FALSE));
    // The URL to the Server Single logout.
    String casSingleLogoutUrl = CommonUtils.constructRedirectUrl(cas
    Properties.getServerUrlPrefix() + "/logout",
        Protocol.CAS2.getServiceParameterName(),
        casSingleLoginUrl,
        Boolean.FALSE, Optional.ofNullable(casProperties.getGate
    way()).orElse(Boolean.FALSE));

    Map<String, Object> result = new HashMap<>(3);
    result.put("code", 200);
    result.put("msg", "");
    result.put("data", "<a href=\"" + casSingleLoginUrl + "\"></a><b
    r><br>"
        + "<a href=\"" + casSingleLogoutUrl + "\"></a>"
    );
    return result;
}

```

```

    }
}

```

- `/cas/index` ☐
- `/cas/login` Cas ☐
- `/cas/logout` Cas ☐
- `/cas/logoutPage` ☐ ☐

2

Yaml

```

#####
#####
###Cas CasClientConfigurationProperties
#####
#####
# CAS-protected client Ignore Pattern Path E.g. /api/cas/**. Optional.
cas-ignore-pattern-path: '**/cas/index,**/cas/logout,**/cas/logoutPage,*
*/swagger-ui/*,**/webjars/*,**/swagger-resources/*,**/v2/api-docs'
# Cas Configuration
cas:
  # CAS server URL E.g. https://example.com/cas or https://cas.example.
  Required.
  server-url-prefix: https://example.com
  # CAS server login URL E.g. https://example.com/cas/login or https://c
  as.example/login. Required.
  server-login-url: ${cas.server-url-prefix}/login
  # CAS-protected client application host URL E.g. https://myclient.exam
  ple.com Required.
  client-host-url: https://myclient.example.com
  # Hostname Verifier to use for HTTPS connections. Defaults to Default
  HostnameVerifier.
  #hostname-verifier: DefaultHostnameVerifier
  # SslConfig for HTTPS connections.
  #ssl-config-file: classpath:conf/cas/ssl.properties
  # ValidationType the CAS protocol validation type. Defaults to CAS3 if
  not explicitly set.
  validation-type: CAS3
  # Validation filter redirectAfterValidation. Defaults to true.
  redirect-after-validation: false
  # Whether to receive the single logout request from cas server.
  single-logout:
    enabled: true

```

Properties

```
# CAS-protected client Ignore Pattern Path E.g. /api/cas/**. Optional.
cas-ignore-pattern-path=**/cas/index,**/cas/logout,**/cas/logoutPage,**/
swagger-ui/**,**/webjars/**,**/swagger-resources/**,**/v2/api-docs
# Cas Configuration
cas.server-url-prefix=https://example.com
cas.server-login-url=${cas.server-url-prefix}/login
cas.client-host-url=https://myclient.example.com
cas.validation-type=CAS3
cas.single-logout.enabled=true
cas.redirect-after-validation=false
```

redirect-after-validation

false ,4

service

3 @EnableCasClient

```
import org.jasig.cas.client.boot.configuration.EnableCasClient;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
@EnableCasClient
public class CasAutoConfigApp {

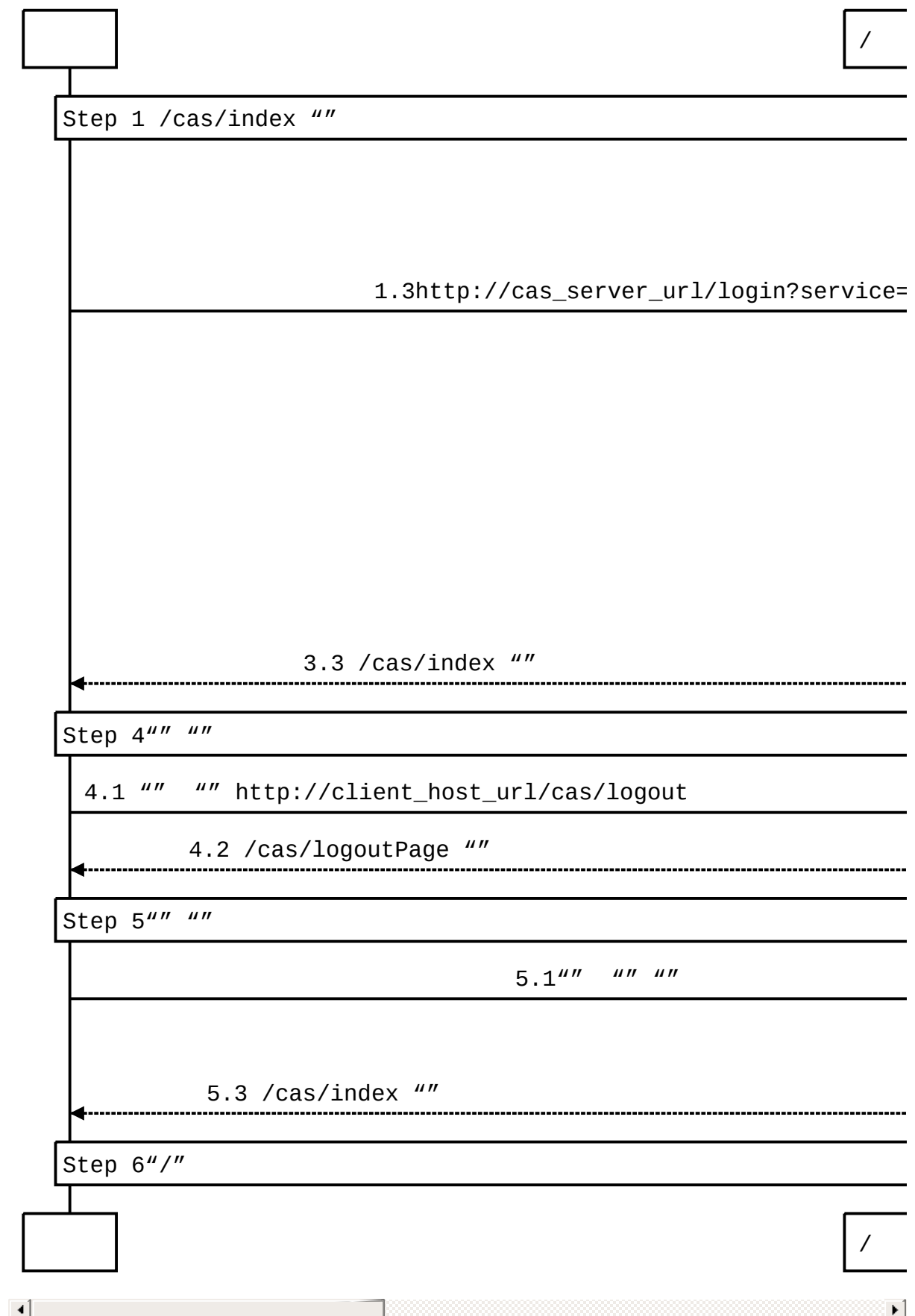
    public static void main(String[] args) {
        SpringApplication.run(CasAutoConfigApp.class, args);
    }

}
```

EnableCasClient CasClientConfiguration,

- casAuthenticationFilter()
- casValidationFilter()
- casHttpServletRequestWrapperFilter()
- casAssertionThreadLocalFilter() Assertion ThreadLocal HttpRequest
- casSingleSignOutFilter()
- casSingleSignOutListener() Listener
- SpringSecurityAssertionAutoConfiguration Spring Security

4Cas





4.1 `/cas/index` , <http://192.168.0.20:8081/cas/index>

- `/cas/index` `/cas/login`
- `/cas/login` Cas `DefaultAuthenticationRedirectStrategy`
- `redirect`

```
public final class DefaultAuthenticationRedirectStrategy implements AuthenticationRedirectStrategy {

    @Override
    public void redirect(final HttpServletRequest request, final HttpServletResponse response,
                        final String potentialRedirectUrl) throws IOException {
        response.sendRedirect(potentialRedirectUrl);
    }
}
```

4.2

- `/cas/login`
- `/cas/login` "" Token Cookie
- `/cas/index` ""

4.3""

```
{
  "code": 200,
  "msg": "",
  "data": "<a href=\"http://192.168.0.20:8081/cas/logout\"></a>"
}
```



- "" "" http://client_host_url/cas/logout
- "" /cas/logoutPage ""

4.4""

```
{
  "code": 200,
  "msg": "",
  "data": "<a href=\"http://192.168.3.27:31495/login?service=http%3A%2F%"
```

```

2F192.168.0.20%3A8081%2Fcas%2Flogin\"></a><br><br><a href=\"http://192.168.3.27:31495/logout?service=http%3A%2F%2F192.168.3.27%3A31495%2Flogin%3Fservice%3Dhttp%253A%252F%252F192.168.0.20%253A8081%252Fcas%252Flogin\">
</a>\"
}

```



- 6677 6677 6677

- /cas/login

- /cas/index ""

Cas

cas/index

cas/logoutPage

1 CustomAuthRedirectStrategy Ajax

Cas casAuthenticationRedirectStrategy401.

CustomAuthRedirectStrategy.java

```

import com.fasterxml.jackson.annotation.JsonAutoDetect;
import com.fasterxml.jackson.annotation.JsonInclude;
import com.fasterxml.jackson.annotation.PropertyAccessor;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.json.JsonMapper;
import lombok.extern.slf4j.Slf4j;
import org.jasig.cas.client.Protocol;
import org.jasig.cas.client.authentication.AuthenticationRedirectStrategy;
import org.jasig.cas.client.boot.configuration.CasClientConfigurationProperties;
import org.jasig.cas.client.util.CommonUtils;
import org.springframework.http.HttpStatus;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.HashMap;
import java.util.Map;
import java.util.Optional;

```

/**

```

    *
    */
@Slf4j
public class CustomAuthRedirectStrategy implements AuthenticationRedirectStrategy {

    public static final String DATE_LONGFORMAT = "yyyy-MM-dd HH:mm:ss";
    private ObjectMapper objectMapper = JsonMapper.builder()
        // finalfinalString,Integer
        // .activateDefaultTyping(LaissezFaireSubTypeValidator.instance, ObjectMapper.DefaultTyping.NON_FINAL)
        .defaultDateFormat(new SimpleDateFormat(DATE_LONGFORMAT))
        // field,getset,ANYprivatepublic
        .visibility(PropertyAccessor.ALL, JsonAutoDetect.Visibility.ANY)
        .serializationInclusion(JsonInclude.Include.NON_NULL).build();

    @Override
    public void redirect(HttpServletRequest request, HttpServletResponse response, String urlToRedirectTo) throws IOException {

        CasClientConfigurationProperties casProperties = SpringBeanUtils
            .getApplicationContext().getBean(CasClientConfigurationProperties.class);
        ;

        String origin = request.getHeader("Origin");
        // Origin
        if (StringUtils.hasText(origin)) {
            response.addHeader("Access-Control-Allow-Origin", origin);
        } else {
            response.addHeader("Access-Control-Allow-Origin", "*");
        }
        //
        response.addHeader("Access-Control-Allow-Headers", "content-type");
        //
        response.addHeader("Access-Control-Allow-Methods", "*");
        // cookie
        response.addHeader("Access-Control-Allow-Credentials", "true");

        if(WebUtils.isAjaxRequest(request)){

            /**
             * The URL to the CAS Server Single login.
             */
            String casSingleLoginUrl = CommonUtils.constructRedirectUrl(

```

```

casProperties.getServerLoginUrl(),
        Protocol.CAS2.getServiceParameterName(),
        casProperties.getClientHostUrl() + "/cas/login",
        Boolean.FALSE, Optional.ofNullable(casProperties.get
Gateway()).orElse(Boolean.FALSE));
    /**
     * The URL to the CAS Server Single logout.
     */
    String casSingleLogoutUrl = CommonUtils.constructRedirectUrl
(casProperties.getServerUrlPrefix() + "/logout",
        Protocol.CAS2.getServiceParameterName(),
        casSingleLoginUrl,
        Boolean.FALSE, Optional.ofNullable(casProperties.get
Gateway()).orElse(Boolean.FALSE));
    if (log.isDebugEnabled()) {
        log.debug("casSingleLoginUrl to \"" + casSingleLoginUrl
+ "\"");
        log.debug("casSingleLogoutUrl to \"" + casSingleLogoutUr
l + "\"");
    }

    response.setStatus(HttpStatus.UNAUTHORIZED.value());
    response.setHeader("content-type", "application/json;charset
=UTF-8");
    response.setCharacterEncoding("UTF-8");

    Map<String, Object> result = new HashMap<>();
    result.put("code", HttpStatus.UNAUTHORIZED.value());
    result.put("msg", "");
    Map<String, Object> data = new HashMap<>();
    data.put("loginUrl", casSingleLoginUrl);
    data.put("logoutUrl", casSingleLogoutUrl);
    result.put("data", data);
    objectMapper.writeValue(response.getOutputStream(), result);

    } else {
        response.sendRedirect(urlToRedirectTo);
    }
}
}

```

2 CustomCasClientConfigurer

```

import org.jasig.cas.client.boot.configuration.CasClientConfigurer;
import org.jasig.cas.client.configuration.ConfigurationKeys;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.boot.web.servlet.FilterRegistrationBean;

```

```

import org.springframework.stereotype.Component;

import java.util.Map;

/**
 * cas-client-support-springboot CAS
 * {
 *      @link CasClientConfigurer}
 */
@Component
public class CustomCasClientConfigurer implements CasClientConfigurer {

    /**
     * CAS-protected client Ignore Pattern Path, E.g: /api/*,/auth/*
     */
    @Value("${cas-ignore-pattern-path}")
    private String ignorePatternPath;

    /**
     * /cas/logout
     */
    @Override
    @SuppressWarnings({ "rawtypes", "unchecked" })
    public void configureAuthenticationFilter(final FilterRegistrationBean authenticationFilter) {
        Map initParameters = authenticationFilter.getInitParameters();
        initParameters.put(ConfigurationKeys.IGNORE_URL_PATTERN_TYPE.getName(), AntUrlPatternMatcherStrategy.class.getName());
        initParameters.put(ConfigurationKeys.AUTHENTICATION_REDIRECT_STRATEGY_CLASS.getName(), CustomAuthRedirectStrategy.class.getName());
        initParameters.put(ConfigurationKeys.IGNORE_PATTERN.getName(), ignorePatternPath);
    }

    @Override
    public void configureValidationFilter(FilterRegistrationBean validationFilter) {
        Map initParameters = validationFilter.getInitParameters();
        initParameters.put(ConfigurationKeys.HOSTNAME_VERIFIER.getName(), TrustAllHostnameVerifier.class.getName());
        initParameters.put(ConfigurationKeys.IGNORE_PATTERN.getName(), ignorePatternPath);
        initParameters.put(ConfigurationKeys.EXCEPTION_ON_VALIDATION_FAILURE.getName(), Boolean.FALSE.toString());
    }
}

```

3 CasAuthController

/cas/address

```

/**
 *
 */
@GetMapping("address")
@ApiOperation("")
public Map<String, Object> getCasAddress() {
    Map<String, Object> result = new HashMap<>(2);
    // The URL to the CAS Server Single login.
    String casSingleLoginUrl = CommonUtils.constructRedirectUrl(casProperties.getServerLoginUrl(),
        Protocol.CAS2.getServiceParameterName(),
        casProperties.getClientHostUrl() + "/cas/login",
        Boolean.FALSE, Optional.ofNullable(casProperties.getGateway())
        .orElse(Boolean.FALSE));
    result.put("loginUrl", casSingleLoginUrl);
    result.put("logoutUrl", casProperties.getClientHostUrl() + "/cas/logout");
    return result;
}

```

4 cas/login

```
cas-client-front-url
```

Yaml

```

# CAS-protected client front url E.g. http://localhost:8080. Required.
cas-client-front-url: http://192.168.0.20:8080

```

Properties

```

# CAS-protected client front url E.g. http://localhost:8080. Required.
cas-client-front-url=http://192.168.0.20:8080

```

```
cas-client-front-url , cas.client-front-url ,
```

CasAuthController casClientFrontUrl login

```

/**
 * CAS-protected client front url E.g. http://localhost:8080. Required.
 */
@Value("${cas-client-front-url}")
private String casClientFrontUrl;

@GetMapping("login")
@ApiOperation("Cas ")

```

```

public void login(HttpServletRequest request, HttpServletResponse response) throws IOException {

    String origin = request.getHeader("Origin");
    // Origin
    if (StringUtils.hasText(origin)) {
        response.addHeader("Access-Control-Allow-Origin", origin);
    } else {
        response.addHeader("Access-Control-Allow-Origin", "");
    }
    //
    response.addHeader("Access-Control-Allow-Headers", "content-type");
    //
    response.addHeader("Access-Control-Allow-Methods", "*");
    // cookie
    response.addHeader("Access-Control-Allow-Credentials", "true");

    // 1CAS
    HttpSession session = request.getSession(false);
    Assertion assertion = (Assertion) (session == null ? request
        .getAttribute(AbstractCasFilter.CONST_CAS_ASSERTION) : session
        .getAttribute(AbstractCasFilter.CONST_CAS_ASSERTION));
    // 2CAS Server
    if (Objects.isNull(assertion)) {
        // 2.1
        String casSingleLoginUrl = CommonUtils.constructRedirectUrl(casProperties.getServerLoginUrl(),
            Protocol.CAS2.getServiceParameterName(),
            casProperties.getClientHostUrl() + "/cas/login",
            Boolean.FALSE, Optional.ofNullable(casProperties.getGateway()).orElse(Boolean.FALSE));
        response.sendRedirect(casSingleLoginUrl);
        return;
    }

    // 3
    Long userId = CasUtil.getUserAccountId();
    Map<String, Object> userInfo = CasUtil.getUserAttributes();
    // 4Cookie
    if (Objects.nonNull(session)) {
        // 5
        final String urlToRedirectTo = casClientFrontUrl + "/home/index?jsessionid=" + session.getId();
        if (log.isDebugEnabled()) {
            log.debug("redirecting to \"\" + urlToRedirectTo + "\"");
        }
        response.sendRedirect(urlToRedirectTo);
    }
}

```



```
}
}
```

```
cas/login jsessionidjsessionidcookiejsessionid
```

```
cas/index
```

5 cas/logout

```
@GetMapping("/logout")
@ApiOperation("cas ")
public void logout(HttpServletRequest request, HttpServletResponse response, @RequestParam(required = false) String url) throws IOException
{
    // 1
    HttpSession session = request.getSession(false);
    if (Objects.nonNull(session)) {
        // 1.1
        session.invalidate();
    }
    // 2CAS ServerURLservice
    // The URL to the CAS Server Single login.
    String casSingleLoginUrl = CommonUtils.constructRedirectUrl(casProperties.getServerLoginUrl(),
        Protocol.CAS2.getServiceParameterName(),
        casProperties.getClientHostUrl() + "/cas/login",
        Boolean.FALSE, Optional.ofNullable(casProperties.getGateway()).orElse(Boolean.FALSE));
    // The URL to the Server Single logout.
    String casSingleLogoutUrl = CommonUtils.constructRedirectUrl(casProperties.getServerUrlPrefix() + "/logout",
        Protocol.CAS2.getServiceParameterName(),
        casSingleLoginUrl,
        Boolean.FALSE, Optional.ofNullable(casProperties.getGateway()).orElse(Boolean.FALSE));
    // 3CAS Server
    response.sendRedirect(casSingleLogoutUrl);
}
```

```
cas/logoutPage
```

6

Yaml

```
#####
```

```
#####
###Cas CasClientConfigurationProperties
#####
#####
# CAS-protected client Ignore Pattern Path E.g. /api/cas/**. Optional.
cas-ignore-pattern-path: '**/cas/address,**/cas/logout,**/swagger-ui/*,*
*/webjars/*,**/swagger-resources/*,**/v2/api-docs'
# Cas Configuration
cas:
  # CAS server URL E.g. https://example.com/cas or https://cas.example.
  Required.
  server-url-prefix: https://example.com
  # CAS server login URL E.g. https://example.com/cas/login or https://c
  as.example/login. Required.
  server-login-url: ${cas.server-url-prefix}/login
  # CAS-protected client application host URL E.g. https://myclient.exam
  ple.com Required.
  client-host-url: https://myclient.example.com
  # Hostname Verifier to use for HTTPS connections. Defaults to Default
  HostnameVerifier.
  #hostname-verifier: DefaultHostnameVerifier
  # SslConfig for HTTPS connections.
  #ssl-config-file: classpath:conf/cas/ssl.properties
  # ValidationType the CAS protocol validation type. Defaults to CAS3 if
  not explicitly set.
  validation-type: CAS3
  # Validation filter redirectAfterValidation. Defaults to true.
  redirect-after-validation: false
  # Whether to receive the single logout request from cas server.
  single-logout:
    enabled: true
```

Properties

```
# CAS-protected client Ignore Pattern Path E.g. /api/cas/**. Optional.
cas-ignore-pattern-path=**/cas/address,**/cas/logout,**/swagger-ui/*,**/
webjars/*,**/swagger-resources/*,**/v2/api-docs
# Cas Configuration
cas.server-url-prefix=https://example.com
cas.server-login-url=${cas.server-url-prefix}/login
cas.client-host-url=https://myclient.example.com
cas.validation-type=CAS3
cas.single-logout.enabled=true
cas.redirect-after-validation=false
```

7Vue

7.1VueRouter JSESSIONID

```

import Router from 'vue-router';
Vue.use(Router);
const vueRouter = new Router({
  mode: 'history',
  base: process.env.VUE_APP_PATH,
  routes: []
});
const beforeEachRoute = async (to, _, next) => {
  const { path, name, query } = to;
  // 1urljsessionId
  if (query.jsessionId) {
    // cookie
    const cookiesList = cookies.keys();
    cookiesList.forEach((item) => {
      cookies.remove(item);
    });
    cookies.set('JSESSIONID', query.jsessionId);
    let _query = { ...query };
    delete _query.jsessionId;
    next({
      path,
      query: _query
    })
    return;
  } else {
    // 2urljsessionId
    const oldJsessionId = cookies.get('JSESSIONID');
    console.log('oldJsessionId', oldJsessionId);
    // 2.1cookieJSESSIONID
    if (oldJsessionId) {
      next();
      return;
    }
    // 2.2cookieJSESSIONID
    getLoginAddress().then((res) => {
      // cookielocalStorage
      const cookiesList = cookies.keys();
      cookiesList.forEach((item) => {
        cookies.remove(item);
      });
      localStorage.removeItem("JSESSIONID");
      //
      console.log('res', res);
      if (res && res.data && res.data.loginUrl) {
        window.open(res.data.loginUrl, '_self');
      }
    })
    .catch(() => {

```

```

        location.reload();
    }).finally(() => {

    });
}
};
vueRouter.beforeEach(beforeEachRoute);

```

7.2Axios HTTP

```

axios.defaults.timeout = 30000;
//
axios.defaults.validateStatus = function (status) {
    return status >= 200 && status <= 500; //
};
// X-Requested-With
axios.defaults.headers["X-Requested-With"] = 'XMLHttpRequest';
// cookie
axios.defaults.withCredentials = true;

```

7.3Axios HTTP 401

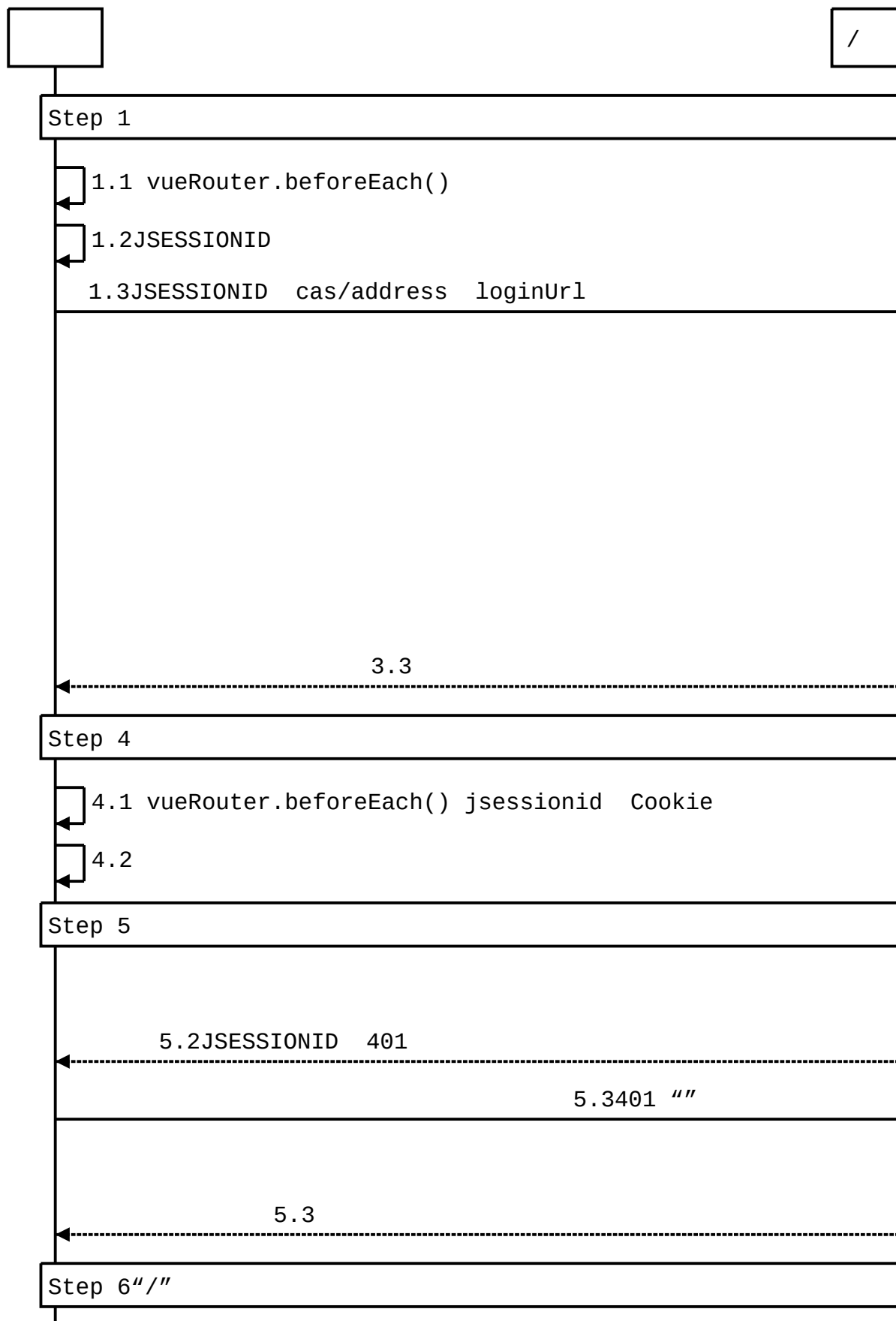
```

import cookies from 'vue-cookies';
import axios from 'axios';

// HTTP
axios.interceptors.response.use((config) => {
    const res = config.data;
    //
    if (config.status === 401 || res.code === 401) {
        // CookieCookie JSESSIONID Session
        const cookiesList = cookies.keys();
        cookiesList.forEach((item) => {
            cookies.remove(item);
        });
        //
        if(res.data && res.data.loginUrl){
            window.open(res.data.loginUrl, '_self');
            return;
        }
    }
    //
}

```

8Cas





8.1 , <http://192.168.0.20:8080>

Vue

- vueRouter.beforeEach()
- JSESSIONID
- JSESSIONID cas/address loginUrl

8.2

-
- "" Token Cookie
-

8.3

- vueRouter.beforeEach() jsessionid Cookie
-
-

8.4

- JSESSIONID
- JSESSIONID 401
- 401 ""
-
-

9

```
CREATE TABLE `ag_user` (
  `id` bigint(19) NOT NULL COMMENT '',
  `account` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT '',
  `is_admin` tinyint(4) DEFAULT NULL COMMENT '',
  `name` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `cas_user_id` bigint(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci ROW_F
```

```
ORMAT=DYNAMIC COMMENT=' ' ;
```

Service

```
import com.baomidou.mybatisplus.extension.service.IService;
import org.jasig.cas.client.validation.Assertion;

public interface IUserService extends IService<User> {

    /**
     * Cas
     * @param assertion
     * @return
     */
    boolean saveOrUpdate(Assertion assertion);

}
```

Service

```
import com.baomidou.mybatisplus.core.conditions.query.LambdaQueryWrapper;
;
import com.baomidou.mybatisplus.core.conditions.update.LambdaUpdateWrapper;
import com.baomidou.mybatisplus.core.toolkit.IdWorker;
import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;
import org.apache.commons.collections4.MapUtils;
import org.jasig.cas.client.validation.Assertion;

import java.util.Map;
import java.util.Objects;

public class UserServiceImpl extends ServiceImpl<UserMapper, User> implements IUserService {

    @Override
    public boolean saveOrUpdate(Assertion assertion) {
        // 1
        Long casUserId = CasUtil.getUserAccountId(assertion);
        Map<String, Object> userInfo = CasUtil.getUserAttributes(assertion);

        // 2casUserId
        User entity = getBaseMapper().selectOne(new LambdaQueryWrapper<User>().eq(User::getCasUserId, casUserId));
        if(Objects.nonNull(entity)){
            entity = new User()
                .setId(IdWorker.getId())
        }
```

```

        .setName(MapUtils.getString(userInfo, "nickname"))
        .setAccount(CasUtil.getUserAccount())
        .setCasUserId(casUserId);
        return this.update(entity, new LambdaUpdateWrapper<User>().eq(
            User::getCasUserId, entity.getCasUserId()));
    } else {
        entity = new User()
            .setId(IdWorker.getId())
            .setName(MapUtils.getString(userInfo, "nickname"))
            .setAccount(CasUtil.getUserAccount())
            .setCasUserId(casUserId)
            .setIsAdmin(Boolean.FALSE);
        return this.save(entity);
    }
}
}
}

```

Mapper

```

@Mapper
public interface UserMapper extends BaseMapper<User> {
}

```

User

```

@Data
@Accessors(chain = true)
@FieldNameConstants
@TableName("ag_user")
public class User {
    @TableId
    private Long id;
    private String account;
    private String name;
    private Boolean isAdmin;
    private Long casUserId;
}

```

/cas/login

```

@GetMapping("login")
@ApiOperation("Cas ")
public void login(HttpServletRequest request, HttpServletResponse response) throws IOException {

    String origin = request.getHeader("Origin");
}

```



```

// Origin
if (StringUtils.hasText(origin)) {
    response.addHeader("Access-Control-Allow-Origin", origin);
} else {
    response.addHeader("Access-Control-Allow-Origin", "*");
}
//
response.addHeader("Access-Control-Allow-Headers", "*");
//
response.addHeader("Access-Control-Allow-Methods", "*");
// cookie
response.addHeader("Access-Control-Allow-Credentials", "true");

// 1CAS
HttpSession session = request.getSession(false);
Assertion assertion = (Assertion) (session == null ? request
    .getAttribute(AbstractCasFilter.CONST_CAS_ASSERTION) : session
    .getAttribute(AbstractCasFilter.CONST_CAS_ASSERTION));
// 2CAS Server
if (Objects.isNull(assertion)) {
    // 2.1
    String casSingleLoginUrl = CommonUtils.constructRedirectUrl(
        casProperties.getServerLoginUrl(),
        Protocol.CAS2.getServiceParameterName(),
        casProperties.getClientHostUrl() + "/cas/login",
        Boolean.FALSE, Optional.ofNullable(casProperties.get
Gateway()).orElse(Boolean.FALSE));
    response.sendRedirect(casSingleLoginUrl);
    return;
}

// 3Cas
userService.saveOrUpdate(assertion);

// 4Cookie
if (Objects.nonNull(session)) {
    // 5
    final String urlToRedirectTo = casClientFrontUrl + "/home/index?jsessionId=" + session.getId();
    if (log.isDebugEnabled()) {
        log.debug("redirecting to \"" + urlToRedirectTo + "\"");
    }
    response.sendRedirect(urlToRedirectTo);
}
}

```

WebUtils.java

```

@Slf4j
public class WebUtils extends org.springframework.web.util.WebUtils {

    private static final String XML_HTTP_REQUEST = "XMLHttpRequest";
    private static final String X_REQUESTED_WITH = "X-Requested-With";
    private static final String CONTENT_TYPE_JSON = "application/json";

    public static boolean isAjaxResponse(HttpServletRequest request ) {
        return isAjaxRequest(request) || isContentTypeJson(request) || isPostRequest(request);
    }

    public static boolean isObjectRequest(HttpServletRequest request ) {
        return isPostRequest(request) && isContentTypeJson(request);
    }

    public static boolean isAjaxRequest(HttpServletRequest request ) {
        return XML_HTTP_REQUEST.equals(request.getHeader(X_REQUESTED_WITH));
    }

    public static boolean isContentTypeJson(HttpServletRequest request ) {
        {
            return request.getHeader(HttpHeaders.CONTENT_TYPE).contains(CONTENT_TYPE_JSON);
        }
    }

    public static boolean isPostRequest(HttpServletRequest request ) {
        return HttpMethod.POST.compareTo(HttpMethod.resolve(request.getMethod())) == 0;
    }

    public static boolean isPostRequest(HttpRequest request ) {
        return HttpMethod.POST.compareTo(request.getMethod()) == 0;
    }
}

```

CasUtil.java

```

import lombok.extern.slf4j.Slf4j;
import org.jasig.cas.client.authentication.AttributePrincipal;

```

```

import org.jasig.cas.client.util.AbstractCasFilter;
import org.jasig.cas.client.util.AssertionHolder;
import org.jasig.cas.client.validation.Assertion;
import org.springframework.web.context.request.RequestContextHolder;
import org.springframework.web.context.request.ServletRequestAttributes;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import java.util.*;

/**
 * Cas
 */
@Slf4j
public class CasUtil {

    /**
     * Assertion
     *
     * @param request request
     * @return Assertion
     */
    public static Assertion getAssertion(HttpServletRequest request) {
        HttpSession session = request.getSession();
        return (Assertion) session.getAttribute(AbstractCasFilter.CONST_
CAS_ASSERTION);
    }

    /**
     * HttpServletRequest
     *
     * @return HttpServletRequest
     */
    public static HttpServletRequest getRequest() {
        ServletRequestAttributes attributes = (ServletRequestAttributes)
RequestContextHolder.getRequestAttributes();
        assert attributes != null;
        return attributes.getRequest();
    }

    /**
     * account
     *
     * @return account
     */
    public static String getUserAccount() {
        log.info("{} ", getRequest().getRemoteUser());
        return getRequest().getRemoteUser();
    }

```

```

    }

    /**
     *
     * @return
     */
    public static Boolean isAdmin() {
        //
        List<String> adminAccounts = getAdminAccountListDateBase();
        //
        return adminAccounts.contains(getUserAccount());
    }

    public static List<String> getAdminAccountListDateBase() {
        return CasUserUtil.getAllAdminAccount();
    }

    /**
     * accountId
     *
     * @return accountId
     */
    public static Long getUserAccountId() {
        return Long.valueOf(String.valueOf(Optional.ofNullable(getUserAttributes().get("accountId")).orElse("0"))));
    }

    /**
     * accountId
     *
     * @return accountId
     */
    public static Long getUserAccountId(Assertion assertion) {
        return Long.valueOf(String.valueOf(Optional.ofNullable(getUserAttributes(assertion).get("accountId")).orElse("0"))));
    }

    /**
     *
     * @return
     */
    public static Map<String, Object> getUserAttributes() {
        Assertion assertion = AssertionHolder.getAssertion();
        return Objects.isNull(assertion) ? Collections.emptyMap() : getUserAttributes(assertion);
    }

```

```

    /**
     *
     * @return
     */
    public static Map<String, Object> getUserAttributes(Assertion assertion) {
        AttributePrincipal principal = assertion.getPrincipal();
        return Objects.isNull(principal) ? new HashMap<>() : principal.getAttributes();
    }

    /**
     * accountId
     *
     * @return accountId
     */
    public static Long getUserAccountIdIgnoreEx() {
        return Long.valueOf(String.valueOf(Optional.ofNullable(getUserAttributesIgnoreEx().get("accountId")).orElse("0"))));
    }

    /**
     *
     *
     * @return
     */
    public static Map<String, Object> getUserAttributesIgnoreEx() {
        try {
            Assertion assertion = AssertionHolder.getAssertion();
            AttributePrincipal principal = assertion.getPrincipal();
            log.info("Principal Name : {} , Attributes: {}", principal.getName(), principal.getAttributes());
            return Objects.isNull(principal) ? Collections.EMPTY_MAP : principal.getAttributes();
        } catch (Exception e){
            log.error(e.getMessage());
            return Collections.EMPTY_MAP;
        }
    }
}

```

SpringBeanUtils.java

```

import org.springframework.context.ApplicationContext;
import org.springframework.context.ApplicationContextAware;

```

```

import org.springframework.stereotype.Component;

@Component
public class SpringBeanUtils implements ApplicationContextAware {

    private static ApplicationContext applicationContext;

    @Override
    public void setApplicationContext(ApplicationContext applicationCont
ext){
        SpringBeanUtils.applicationContext = applicationContext;
    }

    public static ApplicationContext getApplicationContext() {
        return applicationContext;
    }
}

```

1Cas

```

/**
 * The URL to the CAS Server Single login.
 */
String casSingleLoginUrl = CommonUtils.constructRedirectUrl(casPropertie
s.getServerLoginUrl(),
Protocol.CAS2.getServiceParameterName(),
casProperties.getClientHostUrl() + "/cas/login",
Boolean.FALSE, Optional.ofNullable(casProperties.getGateway()).orElse(Bo
olean.FALSE));

/**
 * The URL to the CAS Server Single logout.
 */
String casSingleLogoutUrl = CommonUtils.constructRedirectUrl(casProperti
es.getServerUrlPrefix() + "/logout",
Protocol.CAS2.getServiceParameterName(),
casProperties.getClientHostUrl() + "/cas/logoutPage",
Boolean.FALSE, Optional.ofNullable(casProperties.getGateway()).orElse(Bo
olean.FALSE));

```

- <http://192.168.3.27:31495/login?>

service=http%3A%2F%2F192.168.0.20%3A8081%2Fcas%2Flogin

- <http://192.168.3.27:31495/logout?>

service=http%3A%2F%2F192.168.0.20%3A8081%2Fcas%2FlogoutPage

/cas/logoutPage

AbstractTicketValidationFilter

doFilter

ticket

Rquest

Session

constructServiceUrl

2

<http://192.168.3.27:31495/login?service=http%3A%2F%2F192.168.0.20%3A8081%2Fcas%2Flogin>

doFilter

```
final Assertion assertion = this.ticketValidator.validate(ticket,
    constructServiceUrl(request, response));

logger.debug("Successfully authenticated user: {}", assertion.getPrincipal().getName());

request.setAttribute(CONST_CAS_ASSERTION, assertion);

if (this.useSession) {
    request.getSession().setAttribute(CONST_CAS_ASSERTION, assertion);
}
onSuccessfulValidation(request, response, assertion);

if (this.redirectAfterValidation) {
    logger.debug("Redirecting after successful ticket validation.");
    response.sendRedirect(constructServiceUrl(request, response));
    return;
}
```

constructServiceUrl

```
protected final String constructServiceUrl(final HttpServletRequest request, final HttpServletResponse response) {
    return CommonUtils.constructServiceUrl(request, response, this.service, this.serverName,
        this.protocol.getServiceParameterName(),
        this.protocol.getArtifactParameterName(), this.encodeServiceUrl);
}
```

2

```
request.getRemoteUser();
```

```
AttributePrincipal principal = (AttributePrincipal)request.getUserPrinci  
pal();  
Map attributes = principal.getAttributes();  
Object moblie=attributes .get("moblie");
```

<https://blog.csdn.net/qq2456939181/article/details/127935967>

2.2.3Spring Boot

Spring Boot Cas

1

cas-client-core

Maven

```
<!-- https://mvnrepository.com/artifact/org.jasig.cas.client/cas-client-core -->
<dependency>
    <groupId>org.jasig.cas.client</groupId>
    <artifactId>cas-client-core</artifactId>
    <version>3.6.4</version>
</dependency>
```

Gradle

```
// https://mvnrepository.com/artifact/org.jasig.cas.client/cas-client-core
implementation group: 'org.jasig.cas.client', name: 'cas-client-core', version: '3.6.4'
```

2

1Cas

CasClientCustomProperties

```
import lombok.Data;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.boot.context.properties.ConfigurationProperties;
import org.springframework.cloud.context.config.annotation.RefreshScope;

/**
 * CAS
 */
@ConfigurationProperties(prefix = CasClientProperties.PREFIX)
@Data
@RefreshScope
public class CasClientProperties {
```

```

public static final String PREFIX = "cas";

/**
 * CAS-protected client Web Page host URL
 * E.g. https://myclient.example.com Required.
 */
@Value("${cas.client-host-proxy-url}")
private String clientHostProxyUrl;

/**
 * CAS-protected client Login Api URL
 * E.g: https://myclient.example.com/cas/login
 */
@Value("${cas.client-login-url}")
private String clientLoginUrl;

/**
 * CAS-protected client Logout Api URL
 * E.g: https://myclient.example.com/cas/logout
 */
@Value("${cas.client-logout-url}")
private String clientLogoutUrl;

/**
 * CAS-protected client Ignore Pattern Path
 * E.g: /api/*|/auth/*
 */
@Value("${cas.ignore-pattern-path}")
private String ignorePatternPath;
}

```

```

/**
 *
 */
public class CASProperty {
    /**
     * cas server https://cas_server_url
     */
    private String serverUrlPrefix;
    /**
     * cas server https://cas_server_url/login
     */
    private String serverLoginUrl;
    /**
     * http://cas_client_url

```

```

    */
    private String clientHostUrl;
    /**
     * https://cas_server_url/logout?service=http://client.xxxxx.com/
    casLogin
    */
    private String clientLogoutUrl;
    /**
     * url /api/*|/auth/*
    */
    private String ignorePatternPath;
}

```

```

@Bean
public ServletListenerRegistrationBean<SingleSignOutHttpSessionListe
ner> singleSignOutHttpSessionListener() {
    ServletListenerRegistrationBean<SingleSignOutHttpSessionListe
ner> listener = new ServletListenerRegistrationBean<>();
    listener.setListener(new SingleSignOutHttpSessionListener());
    listener.setOrder(1);
    return listener;
}

/**
 * filter
 * @return
 */
@Bean
public FilterRegistrationBean singleSignOutFilter() {
    FilterRegistrationBean filterRegistration = new FilterRegistrati
onBean();
    filterRegistration.setFilter(new SingleSignOutFilter());
    filterRegistration.addUrlPatterns("/"); filterRegistration.addI
nitParameter("casServerUrlPrefix", casProperty.getServerUrlPrefix());
    filterRegistration.setOrder(1);
    return filterRegistration;
}

/**
 *
 * @return
 */
@Bean
public FilterRegistrationBean authenticationFilter() {
    FilterRegistrationBean filterRegistration = new FilterRegistrati

```

```

onBean();
    filterRegistration.setFilter(new AuthenticationFilter());
    filterRegistration.addUrlPatterns("/");
    //url
    filterRegistration.addInitParameter("ignorePattern", casProperty.getIgnorePatternPath());
    filterRegistration.addInitParameter("casServerLoginUrl", casProperty.getServerLoginUrl());
    filterRegistration.addInitParameter("serverName", casProperty.getClientHostUrl());
    filterRegistration.addInitParameter("useSession", "true");
    filterRegistration.addInitParameter("redirectAfterValidation", "true");
    filterRegistration.setOrder(2);
    return filterRegistration;
}

/**
 * Ticket, CAS 3.0
 * @return
 */
@Bean
public FilterRegistrationBean cas30ProxyReceivingTicketValidationFilter() {
    FilterRegistrationBean filterRegistration = new FilterRegistrationBean();
    filterRegistration.setFilter(new Cas30ProxyReceivingTicketValidationFilter());
    filterRegistration.addUrlPatterns("/");
    filterRegistration.addInitParameter("casServerUrlPrefix", casProperty.getServerUrlPrefix());
    filterRegistration.addInitParameter("serverName", casProperty.getClientHostUrl());
    filterRegistration.setOrder(3);
    return filterRegistration;
}

/**
 * requestrequest.getRemoteUser()
 *
 * @return
 */
@Bean
public FilterRegistrationBean httpServletRequestWrapperFilter() {
    FilterRegistrationBean filterRegistration = new FilterRegistrationBean();
    filterRegistration.setFilter(new HttpServletRequestWrapperFilter());
    filterRegistration.addUrlPatterns("/");
    filterRegistration.setOrder(4);
    return filterRegistration;
}

```

```

@Bean
public FilterRegistrationBean assertionThreadLocalFilter() {
    FilterRegistrationBean filterRegistration = new FilterRegistrationBean();
    filterRegistration.setFilter(new AssertionThreadLocalFilter());
    filterRegistration.addUrlPatterns("/");
    filterRegistration.setOrder(5);
    return filterRegistration;
}

```

id spring boot + spring mvc + vue jsessionId

```

@GetMapping("/casLogin")
@ApiOperation(value = "")
public void login(HttpServletRequest request, HttpServletResponse response) throws IOException {
    HttpSession session = request.getSession();
    Assertion assertion = (Assertion) session.getAttribute(AbstractCasFilter.CONST_CAS_ASSERTION);
    if (assertion != null) {
        String jsessionId = request.getSession().getId();
        response.sendRedirect(casProperty.getFrontUrl()+"?jsessionid="+jsessionid);
    }
}

```

1 casLoginjsessionidjsessionidcookiejsessionid

2 CustomAuthenticationFilter 401

```

@Bean
public FilterRegistrationBean authenticationFilter() {
    FilterRegistrationBean filterRegistration = new FilterRegistrationBean();
    filterRegistration.setFilter(new CustomAuthenticationFilter());
    filterRegistration.addUrlPatterns("/");
    filterRegistration.addInitParameter("ignorePattern", casProperty.getIgnorePatternPath());
    filterRegistration.addInitParameter("casServerLoginUrl", casProperty.getServerLoginUrl());
    filterRegistration.addInitParameter("serverName", casProperty.getClientHostUrl());
    filterRegistration.addInitParameter("useSession", "true");
    filterRegistration.addInitParameter("redirectAfterValidation", "

```

```

true");
    filterRegistration.setOrder(4);
    return filterRegistration;
}

```

```

class CustomAuthenticationFilter extends AbstractCasFilter
doFilter()

```

401

cas server

```

@ApiOperation(value = "" )
@GetMapping("casLogout")
public String logout(HttpServletRequest request) {
    request.getSession().invalidate();
    return "redirect:"+casProperty.getClientLogoutUrl();
}

```

```

request.getRemoteUser();

```

```

AttributePrincipal principal = (AttributePrincipal)request.getUserPrinci
pal();
Map attributes = principal.getAttributes();
Object moblie=attributes .get("moblie");

```

2.2shiro-casShiro

shiro + cas-client-core Cas Spring Boot

Shiro shiro-cas-spring-boot-starter Shiro

1

shiro-cas-spring-boot-starter

Maven

```
<!-- https://mvnrepository.com/artifact/com.github.hiwepy/shiro-cas-spring-boot-starter -->
<dependency>
  <groupId>com.github.hiwepy</groupId>
  <artifactId>shiro-cas-spring-boot-starter</artifactId>
  <version>1.0.5.RELEASE</version>
</dependency>
```

Gradle

```
// https://mvnrepository.com/artifact/com.github.hiwepy/shiro-cas-spring-boot-starter
implementation group: 'com.github.hiwepy', name: 'shiro-cas-spring-boot-starter', version: '1.0.5.RELEASE'
```

2

cas

```
#####
#####
###Shiro
#####
#####
shiro:
  annotations:
    enabled: true
    proxy-target-class: true
  authentication-caching-enabled: false
  authentication-cache-name: SHIRO-AUTHC
  authorization-caching-enabled: false
  authorization-cache-name: SHIRO-AUTHZ
  caching-enabled: false
```

```

cache:
  type: ehcache
enabled: true
kaptcha:
  enabled: true
  retry-times-when-access-denied: 3
failure-url: /error
http:
  header:
    access-control-allow-methods: PUT,POST,GET,DELETE,OPTIONS
login-url: /authz/login/slogin
redirect-url: /authz/login/index
success-url: /index
session-creation-enabled: false
session-validation-scheduler-enabled: false
session-validation-interval: 20000
session-stateless: true
session-storage-enabled: false
session-timeout: 1800000
unauthorized-url: /error
user-native-session-manager: false
web:
  enabled: true
filter-chain-definition-map:
  '['/' : anon
  '['/**/favicon.ico]' : anon
  '['/webjars/**]' : anon
  '['/assets/**]' : anon
  '['/error*]' : anon
  '['/logo/**]' : anon
  '['/swagger-ui.html**]' : anon
  '['/swagger-resources/**]' : anon
  '['/v2/**]' : anon
  '['/kaptcha*]' : anon
  '['/admin]' : anon
  '['/admin/assets/**]' : anon
  '['/admin/applications]' : anon
  '['/admin/applications/**]' : anon
  '['/admin/notifications]' : anon
  '['/admin/notifications/**]' : anon
  '['/admin/instances]' : anon
  '['/admin/instances/**]' : anon
  '['/sockets/**]' : anon
  '['/expiry]' : cros,withinExpiry
  '['/authz/login/slogin]' : cros,authc
  '['/logout]' : logout
cas:
  accept-any-proxy: true

```



```

cas-server-login-url: http://127.0.0.1:10000/cas/login
cas-server-logout-url: http://127.0.0.1:10000/cas/logout
cas-server-url-prefix: http://127.0.0.1:10000/cas
enabled: true
encoding: UTF-8
server-callback-url: /callback
server-name: http://127.0.0.1:8080
ignore-pattern: /webjars/;/assets/;/authz/login;/logout;/callback
ignore-url-pattern-type: org.apache.shiro.spring.boot.cas.ContainsPa
tternsUrlPatternMatcherStrategy

```

3 ShrioFilterChainDefinitionConfigurer

```

import org.apache.shiro.spring.boot.FilterChainDefinitionConfigurer;
import org.apache.shiro.spring.web.config.DefaultShiroFilterChainDefinit
ion;
import org.springframework.stereotype.Component;

@Component
public class ShrioFilterChainDefinitionConfigurer implements FilterChain
DefinitionConfigurer {

    @Override
    public void configurePathDefinition(DefaultShiroFilterChainDefinitio
n chainDefinition) {
        chainDefinition.addPathDefinition("/*", "headers,cas");
        chainDefinition.addPathDefinition("/**", "headers,cas");
    }
}

```

2.3spring-security-casSpring Security

spring-security-cas Cas Spring Boot

spring-security-cas Spring Security cas-client-core Spring Security

1

spring-security-cas

Maven

```
<!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-
boot-starter-security -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
    <version>2.7.2</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.springframework.security/spr
ing-security-cas -->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-cas</artifactId>
    <version>5.7.2</version>
</dependency>
```

Gradle

```
// https://mvnrepository.com/artifact/org.springframework.boot/spring-bo
ot-starter-security
implementation group: 'org.springframework.boot', name: 'spring-boot-sta
rter-security', version: '2.7.2'
// https://mvnrepository.com/artifact/org.springframework.security/sprin
g-security-cas
implementation group: 'org.springframework.security', name: 'spring-secu
rity-cas', version: '5.7.2'
```

2

YmlProperties application.yml

```
@Component
@Data
public class YmlProperties {
```

```

@Value("${app.server.host.url}")
private String appServerUrl;

@Value("${app.login.url}")
private String appLoginUrl;

@Value("${app.logout.url}")
private String appLogoutUrl;

@Value("${cas.server.host}")
private String casServerUrl;

@Value("${cas.server.login_url}")
private String casServerLoginUrl;

@Value("${cas.server.logout_url}")
private String casServerLogoutUrl;
}

```

UserDetail

```
spring securityorg.springframework.security.core.userdetails.User
```

```

public class UserDetails extends User implements UserDetails, Credentials
Container {

    private Long userId;

    public UserDetails(String username,String password, Collection<? extend
s GrantedAuthority> authorities){
        super(username,password,true,true,true,true,authorities);
    }

    public Long getUserId() {
        return userId;
    }

    public void setUserId(Long userId) {
        this.userId = userId;
    }
}

```

MyUserDetailsService AuthenticationUserDetailsService

Cas

```

@Service
public class MyUserDetailsService implements AuthenticationUserDetailsService<CasAssertionAuthenticationToken>{

    @Autowired
    private UserService userService;

    @Autowired
    private PermissionMapper permissionMapper;

    @Override
    public UserDetails loadUserDetails(CasAssertionAuthenticationToken token)
        throws UsernameNotFoundException {
        User user = userService.findByName(token.getName());
        if (Objects.isNull(user)) {
            throw new RuntimeException(" ");
        }
        List<String> roleCodes = permissionMapper.getPermissionByUser(user.getId());
        if (CollectionUtils.isEmpty(roleCodes)) {
            throw new RuntimeException("[ " + user.getUsername() + " ] ");
        }
        Set<GrantedAuthority> authorities = new HashSet<>();
        for (String roleCode : roleCodes) {
            authorities.add(new SimpleGrantedAuthority(roleCode));
        }

        UserDetail userDetail=new UserDetail(user.getUsername(),user.getPassword(),authorities);
        userDetail.setUserId(user.getId());
        return userDetail;
    }
}

```

SecurityCasConfiguration WebSecurityConfigurerAdapter

```

@Slf4j
@EnableWebSecurity
//@Secured@PreAuthorize
@EnableGlobalMethodSecurity(securedEnabled = true, prePostEnabled = true)
@Configuration
public class SecurityCasConfiguration extends WebSecurityConfigurerAdapter

```

```

er {

    @Autowired
    private YmlProperties ymlProperties;

    @Autowired
    private MyUserDetailsService myUserDetailsService;

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests() //
            .antMatchers("/login").permitAll()
            .anyRequest().authenticated()
            .and()
            .csrf().disable()
            .logout().permitAll() //logout
            .and()
            .headers().frameOptions().disable()
            .and()
            .cors()
            .and().formLogin(); //form

        http.exceptionHandling().authenticationEntryPoint(myCasAuthenticationEntryPoint)
            .and()
            .addFilter(casAuthenticationFilter())
            .addFilterBefore(casLogoutFilter(), LogoutFilter.class)
            .addFilterBefore(singleSignOutFilter(), CasAuthenticationFilter.class);
    }

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
        super.configure(auth);
        auth.authenticationProvider(casAuthenticationProvider());
    }

    /**
     * ServicePropertyesservicecas
     */
    @Bean
    public ServiceProperties serviceProperties() {
        ServiceProperties serviceProperties = new ServiceProperties();
        serviceProperties.setService(ymlProperties.getAppServerUrl() + ymlProperties.getAppLoginUrl());
        serviceProperties.setSendRenew(false);
    }

```

```

        serviceProperties.setAuthenticateAllArtifacts(true);
        return serviceProperties;
    }

    /**
     * casServiceProperties()
     */
    @Bean
    public CasAuthenticationEntryPoint casAuthenticationEntryPoint() {
        CasAuthenticationEntryPoint casAuthenticationEntryPoint = new CasAuth
        henticationEntryPoint();
        casAuthenticationEntryPoint
            .setLoginUrl(ymlProperties.getCasServerLoginUrl());
        casAuthenticationEntryPoint.setServiceProperties(serviceProperties()
        );
        return casAuthenticationEntryPoint;
    }

    @Bean
    public CasAuthenticationFilter casAuthenticationFilter() throws Except
    ion {
        CasAuthenticationFilter casAuthenticationFilter = new CasAuthenticat
        ionFilter();
        casAuthenticationFilter.setServiceProperties(serviceProperties());
        casAuthenticationFilter.setFilterProcessesUrl(ymlProperties.getAppLogi
        nUrl());
        casAuthenticationFilter.setAuthenticationManager(authenticationManag
        er());

        //SavedRequestAwareAuthenticationSuccessHandler
        // SavedRequestAwareAuthenticationSuccessHandler handler=new SavedReq
        uestAwareAuthenticationSuccessHandler();
        // handler.setRedirectStrategy(new MyRedirectStrategy());
        // casAuthenticationFilter.setAuthenticationSuccessHandler(handler);
        casAuthenticationFilter.setAuthenticationSuccessHandler(
            new SimpleUrlAuthenticationSuccessHandler(
                ymlProperties.getAppServerUrl() + "/hello"));
        casAuthenticationFilter.setSessionAuthenticationStrategy(sessionAuth
        enticationStrategy());
        return casAuthenticationFilter;
    }

    // class MyRedirectStrategy extends DefaultRedirectStrategy {
    //
    // @Override
    // public void sendRedirect(HttpServletRequest request, HttpServletResponse
    response, String url)

```

```
//      throws IOException {
//      String redirectUrl = calculateRedirectUrl(request.getContextPath
//      (), url);
//      redirectUrl = response.encodeRedirectURL(redirectUrl);
//      log.info(redirectUrl);
//      if(redirectUrl.startsWith("http://")){
//      if(redirectUrl.contains("/app/login")){
//      redirectUrl="http://localhost:8080/test/index";
//      }
//      }
//      response.sendRedirect(redirectUrl);
//      }
//      }
```

@Bean

```
public CasAuthenticationProvider casAuthenticationProvider() {
    CasAuthenticationProvider casAuthenticationProvider = new CasAuthent
icationProvider();
    casAuthenticationProvider.setServiceProperties(serviceProperties());
    casAuthenticationProvider.setTicketValidator(cas20ServiceTicketValid
ator());
    casAuthenticationProvider
        .setAuthenticationUserDetailsService(myUserDetailsService);
    casAuthenticationProvider.setKey("casAuthenticationProviderKey");
    return casAuthenticationProvider;
}
```

/**

*** tickercas**

***/**

@Bean

```
public Cas20ServiceTicketValidator cas20ServiceTicketValidator() {
    //Https
    //    HttpURLConnectionFactory httpURLConnectionFactory=new HttpURLConne
ctionFactory() {
    //        @Override
    //        public HttpURLConnection buildHttpURLConnection(URLConnection ur
lConnection) {
    //            SSLContext sslContext=null;
    //            try {
    //                sslContext = SSLContext.getInstance("SSL");
    //                sslContext.init(new KeyManager[] {}, new TrustManager[] {new X5
09TrustManager() }
    //                @Override
    //                public void checkClientTrusted(X509Certificate[] x509Certi
ficates, String s)
```

```

//          throws CertificateException {
//
//      }
//
//      @Override
//      public void checkServerTrusted(X509Certificate[] x509Certificates, String s)
//          throws CertificateException {
//
//      }
//
//      @Override
//      public X509Certificate[] getAcceptedIssuers() {
//          return new X509Certificate[0];
//      }
//  },null);
//  SSLSocketFactory socketFactory=sslContext.getSocketFactory()
//  ;
//
//      if(urlConnection instanceof HttpURLConnection){
//          HttpURLConnection httpsURLConnection=(HttpURLConnection)
urlConnection;
//          httpsURLConnection.setSSLSocketFactory(socketFactory);
//          httpsURLConnection.setHostnameVerifier((s,l)->true);
//      }
//
//      return (HttpURLConnection)urlConnection;
//
//  }catch (Exception e){
//      throw new RuntimeException(e);
//  }
//  }
//  };
    Cas20ServiceTicketValidator cas20ServiceTicketValidator = new Cas20ServiceTicketValidator(
        ymlProperties.getCasServerUrl());
    cas20ServiceTicketValidator.setEncoding("UTF-8");
    return cas20ServiceTicketValidator;
}

```

```

@Bean
public SessionAuthenticationStrategy sessionAuthenticationStrategy() {
    SessionAuthenticationStrategy sessionAuthenticationStrategy = new SessionFixationProtectionStrategy();
    return sessionAuthenticationStrategy;
}

```



```

/**
 * cas
 */
@Bean
public SingleSignOutFilter singleSignOutFilter() {
    SingleSignOutFilter singleSignOutFilter = new SingleSignOutFilter();
    singleSignOutFilter.setCasServerUrlPrefix(ymlProperties.getCasServer
Url());
    singleSignOutFilter.setIgnoreInitConfiguration(true);
    return singleSignOutFilter;
}

/**
 * logoutlogoutcas
 */
@Bean
public LogoutFilter casLogoutFilter() {
    LogoutFilter logoutFilter = new LogoutFilter(ymlProperties.getCasSer
verLogoutUrl(),
        new SecurityContextLogoutHandler());
    logoutFilter.setFilterProcessesUrl(ymlProperties.getAppLogoutUrl());
    return logoutFilter;
}

/**
 * @Secured "ROLE_"
 * @return
 */
@Bean
public GrantedAuthorityDefaults grantedAuthorityDefaults() {
    return new GrantedAuthorityDefaults("");
}
}

```

HelloController

```

@Slf4j
@RestController
public class HelloController {

    @Secured("security_index")
    // @PreAuthorize("hasAnyAuthority('security_index')")
    @GetMapping("/hello")
    public String hello() {
        return "" ;
    }
}

```

```
}
}
```

<http://localhost:8080/test/login> CAS

```
CAS
AuthenticationEntryPoint
CASCAS
```

HelloController

Resultjson

```
@Slf4j
@RestController
public class HelloController {

    @Autowired
    private YmlProperties ymlProperties;

    /**
     *
     * CAS
     * @return
     */
    @GetMapping("/send")
    public Result send() {
        String url =
            ymlProperties.getCasServerLoginUrl() + "?service=" + ymlProperties.getAppServerUrl()
            + ymlProperties.getAppLoginUrl();
        return Result.failed().setCode(444).setData(url).setMessage("").setSuccess(false);
    }

    /**
     *
     * @return
     */
    @GetMapping("/login")
    public Result login(){
        return Result.success(null, "");
    }
}
```



Resultjson

```
public interface IErrorCode {
    long getCode();
    String getMessage();
}

public enum ResultCode implements IErrorCode {
    FAILED(500, "" ),
    SUCCESS(200, "" );
    private long code;
    private String message;

    private ResultCode(long code,String message){
        this.code=code;
        this.message=message;
    }

    @Override
    public long getCode() {
        return code;
    }

    @Override
    public String getMessage() {
        return message;
    }
}
```

Result

```
@Data
@Accessors(chain = true)
public class Result<T> {
    private boolean success;
    private long code;
    private String message;
    private T data;

    private Result() {
    }

    private Result(boolean success,long code,String message,T data){
        this.success=success;
        this.code=code;
        this.message=message;
        this.data=data;
    }
}
```

```

    }

    /**
     *
     * @param data
     */
    public static <T> Result<T> success(T data){
        return new Result<T>(true, ResultCode.SUCCESS.getCode(), ResultCode
e.SUCCESS.getMessage(), data);
    }

    /**
     *
     * @param data
     * @param message
     */
    public static <T> Result<T> success(T data, String message){
        return new Result<>(true, ResultCode.SUCCESS.getCode(), message, da
ta);
    }

    /**
     *
     * @param <T>
     * @return
     */
    public static <T> Result<T> failed() {
        return new Result<>(false, ResultCode.FAILED.getCode(), ResultCode
.FAILED.getMessage(), null);
    }

    /**
     *
     * @param message
     */
    public static <T> Result<T> failed(String message){
        return new Result<>(false, ResultCode.FAILED.getCode(), message, nu
ll);
    }
}

```

MyCasAuthenticationEntryPointAuthenticationEntryPoint

```

@Component
public class MyCasAuthenticationEntryPoint implements AuthenticationEntr
ypoint {

```

```

@Autowired
private YmlProperties ymlProperties;

@Override
public void commence(HttpServletRequest request, HttpServletResponse response,
    AuthenticationException authException) throws IOException, ServletException {
    response.sendRedirect(ymlProperties.getAppServerUrl()+"/send");
}
}

```

SecurityCasConfiguration

```

@Slf4j
@EnableWebSecurity
//@Secured@PreAuthorize
@EnableGlobalMethodSecurity(securedEnabled = true, prePostEnabled = true)
@Configuration
public class SecurityCasConfiguration extends WebSecurityConfigurerAdapter {

    @Autowired
    private YmlProperties ymlProperties;

    @Autowired
    private MyCasAuthenticationEntryPoint myCasAuthenticationEntryPoint;

    @Autowired
    private MyUserDetailsService myUserDetailsService;

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests() //
            .antMatchers("/login", "/send").permitAll()
            .anyRequest().authenticated() //
            .and()
            .csrf().disable()
            .logout().permitAll() //logout
            .and()
            .headers().frameOptions().disable()
            .and()
            .cors()
            .and().formLogin(); //form

        http.exceptionHandling().authenticationEntryPoint(myCasAuthenticatio

```

```

nEntryPoint)
    .and()
    .addFilter(casAuthenticationFilter())
    .addFilterBefore(casLogoutFilter(), LogoutFilter.class)
    .addFilterBefore(singleSignOutFilter(), CasAuthenticationFilter.
class);
}

```

```

@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exc
eption {
    super.configure(auth);
    auth.authenticationProvider(casAuthenticationProvider());
}

```

```

/**
 * ServiceProperties servicecas
 */
@Bean
public ServiceProperties serviceProperties() {
    ServiceProperties serviceProperties = new ServiceProperties();
    serviceProperties.setService(ymlProperties.getAppServerUrl() + ymlPr
operties.getAppLoginUrl());
    serviceProperties.setSendRenew(false);
    serviceProperties.setAuthenticateAllArtifacts(true);
    return serviceProperties;
}

```

```

/**
 * casServiceProperties()
 */
// @Bean
// public CasAuthenticationEntryPoint casAuthenticationEntryPoint() {
//     CasAuthenticationEntryPoint casAuthenticationEntryPoint = new CasA
uthenticationEntryPoint();
//     casAuthenticationEntryPoint
//         .setLoginUrl(casProperties.getCasServerLoginUrl());
//     casAuthenticationEntryPoint.setServiceProperties(serviceProperties
());
//     return casAuthenticationEntryPoint;
// }

```

```

@Bean
public CasAuthenticationFilter casAuthenticationFilter() throws Except
ion {
    CasAuthenticationFilter casAuthenticationFilter = new CasAuthenticat

```

```

ionFilter();
    casAuthenticationFilter.setServiceProperties(serviceProperties());
    casAuthenticationFilter.setFilterProcessesUrl(ymProperties.getAppLo
ginUrl());
    casAuthenticationFilter.setAuthenticationManager(authenticationManag
er());

    //SavedRequestAwareAuthenticationSuccessHandler
    //    SavedRequestAwareAuthenticationSuccessHandler handler=new SavedReq
uestAwareAuthenticationSuccessHandler();
    //    handler.setRedirectStrategy(new MyRedirectStrategy());
    //    casAuthenticationFilter.setAuthenticationSuccessHandler(handler);
    casAuthenticationFilter.setAuthenticationSuccessHandler(
        new SimpleUrlAuthenticationSuccessHandler(
            ymlProperties.getAppServerUrl() + "/hello"));
    casAuthenticationFilter.setSessionAuthenticationStrategy(sessionAuth
enticationStrategy());
    return casAuthenticationFilter;
}

```

```

// class MyRedirectStrategy extends DefaultRedirectStrategy {
//
//    @Override
//    public void sendRedirect(HttpServletRequest request, HttpServletResponse
response response, String url)
//        throws IOException {
//        String redirectUrl = calculateRedirectUrl(request.getContextPath
()), url);
//        redirectUrl = response.encodeRedirectURL(redirectUrl);
//        log.info(redirectUrl);
//        if(redirectUrl.startsWith("http://")){
//            if(redirectUrl.contains("/app/login")){
//                redirectUrl="http://10.11.36.21:8080/test/hello";
//            }
//        }
//        response.sendRedirect(redirectUrl);
//    }
// }

```

```

@Bean
public CasAuthenticationProvider casAuthenticationProvider() {
    CasAuthenticationProvider casAuthenticationProvider = new CasAuthent
icationProvider();
    casAuthenticationProvider.setServiceProperties(serviceProperties());
    casAuthenticationProvider.setTicketValidator(cas20ServiceTicketValid
ator());
    casAuthenticationProvider

```

```

        .setAuthenticationUserDetailsService(myUserDetailsService);
        casAuthenticationProvider.setKey("casAuthenticationProviderKey");
        return casAuthenticationProvider;
    }

    /**
     * tickercas
     */
    @Bean
    public Cas20ServiceTicketValidator cas20ServiceTicketValidator() {
        //Https
        //    HttpURLConnectionFactory httpURLConnectionFactory=new HttpURLConne
        ctionFactory() {
        //        @Override
        //        public HttpURLConnection buildHttpURLConnection(URLConnection ur
        lConnection) {
        //            SSLContext sslContext=null;
        //            try {
        //                sslContext = SSLContext.getInstance("SSL");
        //                sslContext.init(new KeyManager[] {}, new TrustManager[] {new X5
        09TrustManager() {
        //                    @Override
        //                    public void checkClientTrusted(X509Certificate[] x509Certi
        ficates, String s)
        //                        throws CertificateException {
        //
        //
        //                    }
        //
        //                @Override
        //                public void checkServerTrusted(X509Certificate[] x509Certi
        ficates, String s)
        //                    throws CertificateException {
        //
        //
        //                }
        //
        //                @Override
        //                public X509Certificate[] getAcceptedIssuers() {
        //                    return new X509Certificate[0];
        //                }
        //            }}, null);
        //            SSLSocketFactory socketFactory=sslContext.getSocketFactory()
        ;
        //
        //            if(urlConnection instanceof HttpsURLConnection){
        //                HttpsURLConnection httpsURLConnection=(HttpsURLConnection)
        urlConnection;
        //                httpsURLConnection.setSSLSocketFactory(socketFactory);
    
```



```

//          httpsURLConnection.setHostnameVerifier((s,l)->true);
//      }
//
//          return (HttpURLConnection)urlConnection;
//
//      }catch (Exception e){
//          throw new RuntimeException(e);
//      }
//  }
//  };

    Cas20ServiceTicketValidator cas20ServiceTicketValidator = new Cas20S
erviceTicketValidator(
        ymlProperties.getCasServerUrl());
    cas20ServiceTicketValidator.setEncoding("UTF-8");
    return cas20ServiceTicketValidator;
}

@Bean
public SessionAuthenticationStrategy sessionAuthenticationStrategy() {
    SessionAuthenticationStrategy sessionAuthenticationStrategy = new Se
ssionFixationProtectionStrategy();
    return sessionAuthenticationStrategy;
}

/**
 * cas
 */
@Bean
public SingleSignOutFilter singleSignOutFilter() {
    SingleSignOutFilter singleSignOutFilter = new SingleSignOutFilter();
    singleSignOutFilter.setCasServerUrlPrefix(ymlProperties.getCasServer
Url());
    singleSignOutFilter.setIgnoreInitConfiguration(true);
    return singleSignOutFilter;
}

/**
 * logoutlogoutcas
 */
@Bean
public LogoutFilter casLogoutFilter() {
    LogoutFilter logoutFilter = new LogoutFilter(ymlProperties.getCasSer
verLogoutUrl(),
        new SecurityContextLogoutHandler());
    logoutFilter.setFilterProcessesUrl(ymlProperties.getAppLogoutUrl());
    return logoutFilter;
}

```

```

    }

    /**
     * @Secured "ROLE_"
     * @return
     */
    @Bean
    public GrantedAuthorityDefaults grantedAuthorityDefaults() {
        return new GrantedAuthorityDefaults("");
    }
}

```

/login 200"/send444CAS444CAS

:

https://blog.csdn.net/new_ord/article/details/109709181

3.1

OpenID

1

2OpenID Connect

3OpenID

OpenID Connect

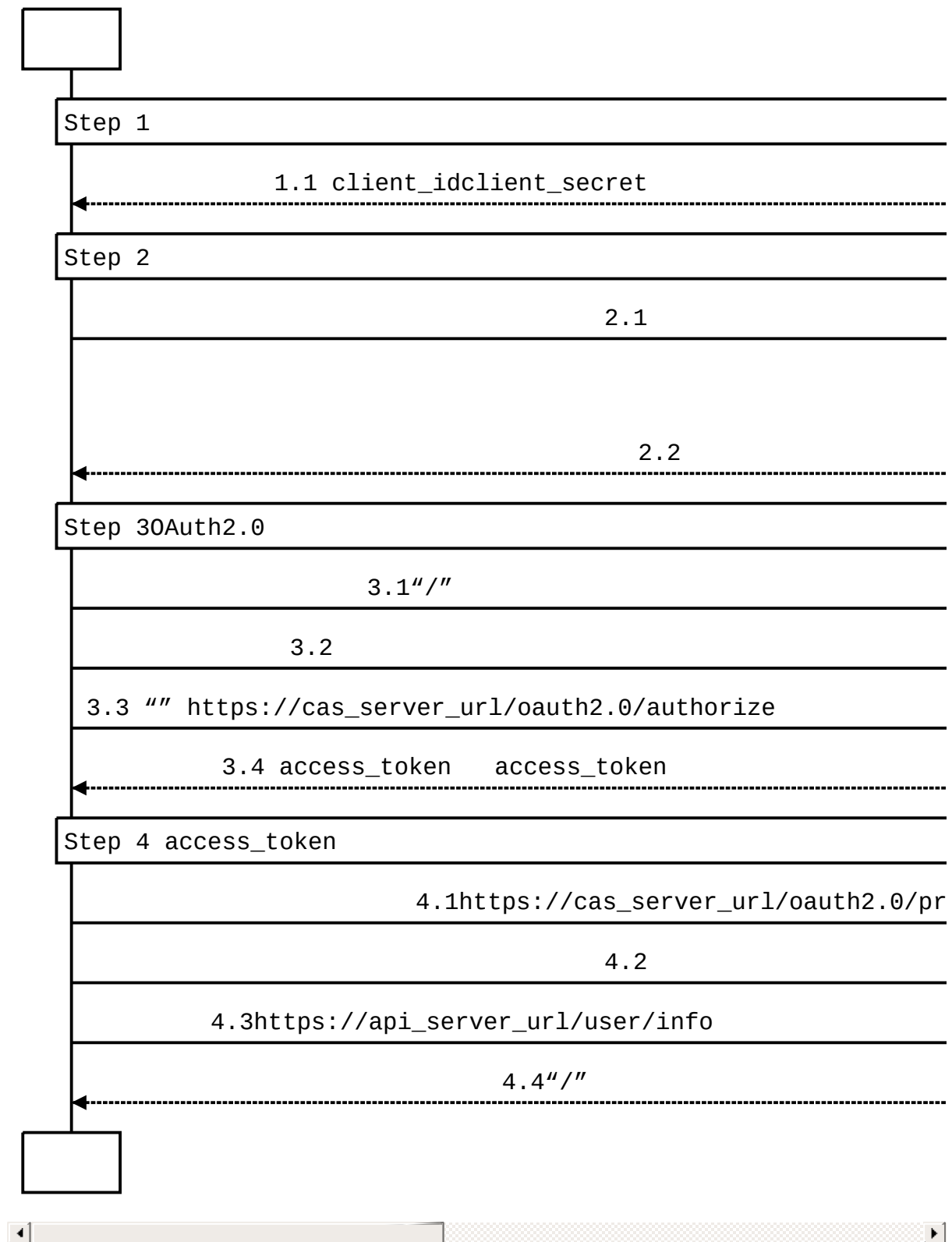
OAuth2.0

1

OAuth2.0

OAuth2.0 - B/S accessToken

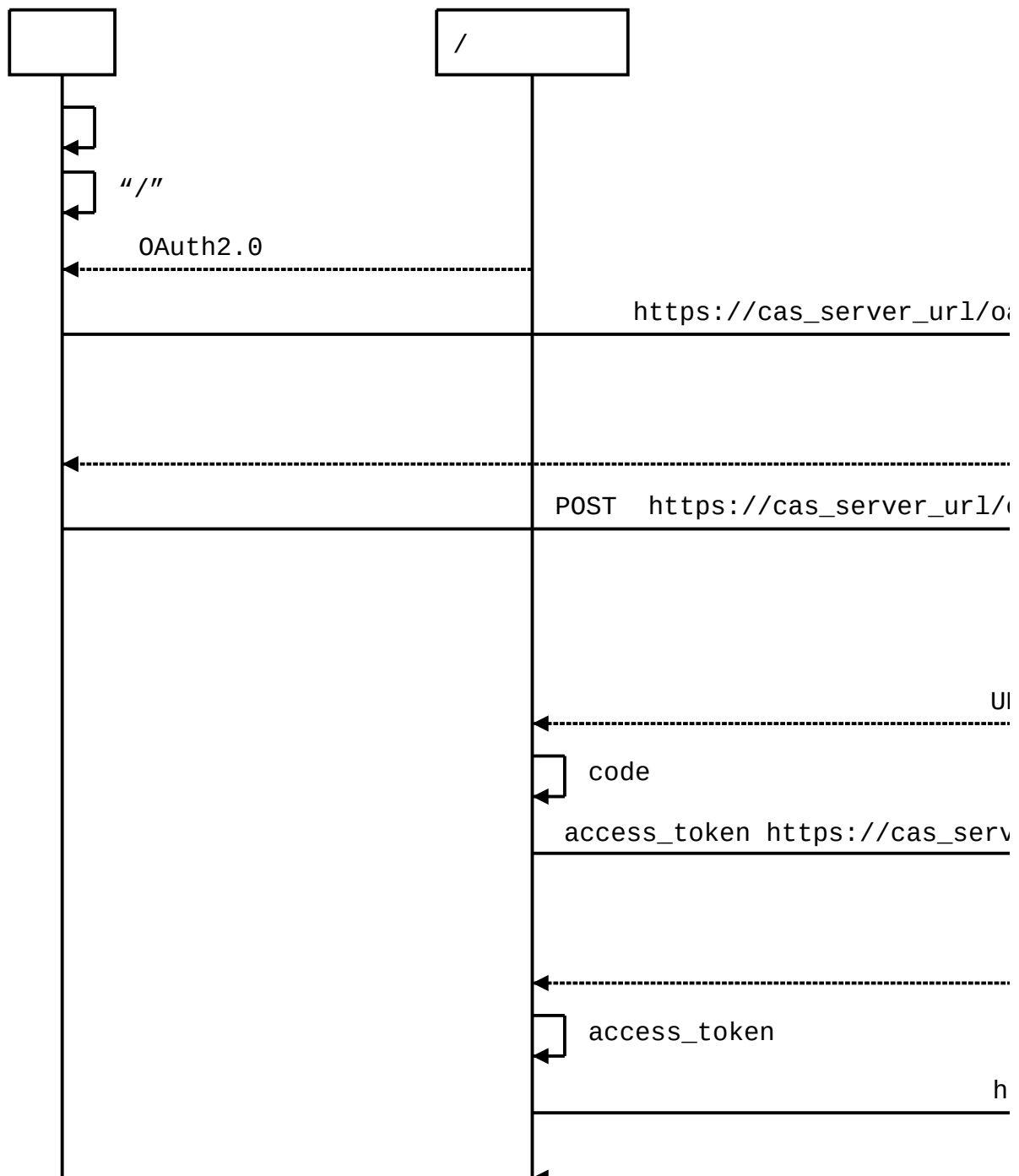
- Authorization CodeaccessToken“/”
- ImplicitaccessToken“/”
- Password Credentials“/”
- Client Credentials“/”

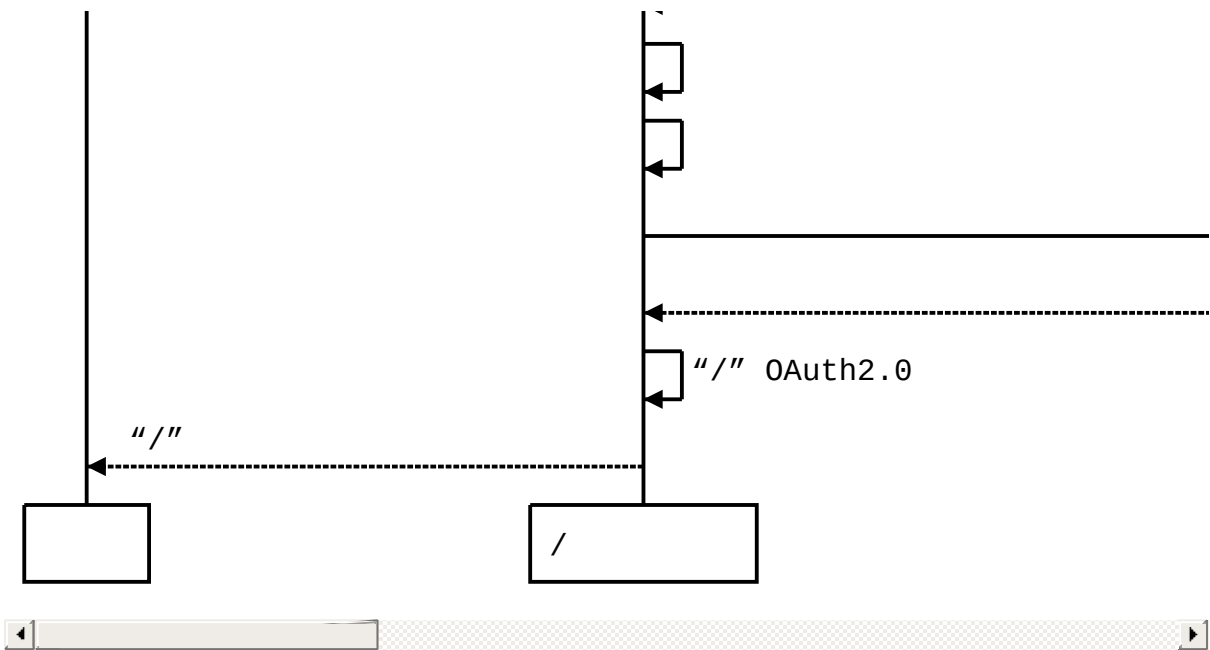


2Authorization Code

OAuth2.0 Authorization Code

Authorization CodeAPPcode"/"





code

```
OAuth2.0 https://cas_server_url/oauth2.0/authorize?
response_type=code&client_id=yourclient&redirect_uri=https://my_server_url/callback
```

					schema
client_id	client_id	query	true	string	
response_type	code	query	true	string	
redirect_uri		query	true	string	
state	statecsrfsession	query	false	string	

OAuth2.0 code

```
response.sendRedirect("https://cas_server_url/oauth2.0/authorize?response_type=code&client_id=yourclient&redirect_uri=https://my_server_url/callback");
```

redirect_uricodestate redirect_uri?code=CODE&state=STATE

```
redirect_urihttpscode
```

code access_token

code access_token [https://cas_server_url/oauth2.0/accessToken?
grant_type=authorization_code&client_id=yourclient&client_secret=yoursecret&code=yourcode
&redirect_uri=https://my_server_url/callback](https://cas_server_url/oauth2.0/accessToken?grant_type=authorization_code&client_id=yourclient&client_secret=yoursecret&code=yourcode&redirect_uri=https://my_server_url/callback)

					schema
client_id	client_id	body	true	string	
client_secret	client_secret	body	true	string	
grant_type	authorization_code	body	true	string	
code		body	true	string	
redirect_uri		body	true	string	

```
{
  "service" : "service_url",
  "attributes" : {
    "credentialType" : "LoginCredential"
  },
  "id" : "admin",
  "client_id" : "client"
}
```

access_token

accesstoken https://cas_server_url/oauth2.0/profile?access_token=xxxxx

```
{
  "service" : "service_url",
  "attributes" : {
    "credentialType" : "LoginCredential"
  },
  "id" : "admin",
  "client_id" : "client"
}
```

			schema
service		string	
attributes		integer(int64)	
id	ID	string	
client_id	ID	string	

code

code

code

:

- https://cas_server_url/oauth2.0/authorize

https://cas_server_url/oauth2.0/authorize? response_type=code&client_id=yourclient&redirect_uri=https://my_server_url/casLogin
--

- GET

					schema

:

					schema
client_id	client_id	query	true	string	
response_type	code	query	true	string	
redirect_uri		query	true	string	
state	statecsrfsession	query	false	string	

		schema
--	--	--------

code

200	OK	
-----	----	--

redirect_uricodestateredirect_uri?code=CODE&state=STATE

redirect_urihttpscode

access_token

code access_token

code access_token

:

- https://cas_server_url/oauth2.0/accessToken

[https://cas_server_url/oauth2.0/accessToken?
grant_type=authorization_code&client_id=yourclient&client_secret=yoursecret&code=yourcode
&redirect_uri=https://my_server_url/casLogin](https://cas_server_url/oauth2.0/accessToken?grant_type=authorization_code&client_id=yourclient&client_secret=yoursecret&code=yourcode&redirect_uri=https://my_server_url/casLogin)

- POST

					schema

:

					schema
client_id	client_id	body	true	string	
client_secret	client_secret	body	true	string	
grant_type	authorization_code	body	true	string	
code		body	true	string	
redirect_uri		body	true	string	

		schema
200	OK	

:

:

```
{
  "access_token": "AT-3-rX4Wf0iCgiF6qMnt52PfQDaQj-F4I9LK",
  "token_type": "bearer",
  "expires_in": 28800
}
```

:

```
{
  "timestamp": 1660987198227,
  "status": 401,
  "error": "Unauthorized",
  "message": "No message available",
  "path": "/oauth2.0/accessToken"
}
```

:

			schema
access_token	access_token	string	
token_type	bearer	string	
expires_in		integer(int32)	

:

JavaScript - Fetch

```
var myHeaders = new Headers();
myHeaders.append("service", "http://192.168.3.25:8090");
myHeaders.append("Content-Type", "application/json");

var requestOptions = {
```

```
access_token
```

```
method: 'POST',  
headers: myHeaders,  
redirect: 'follow'  
};  
  
fetch("https://cas_server_url/oauth2.0/accessToken?grant_type=authorization_code&client_secret=xxx=&username=xxx&password=xxx&client_id=xxx", requestOptions)  
  .then(response => response.text())  
  .then(result => console.log(result))  
  .catch(error => console.log('error', error));
```

Java - OkHttp

```
OkHttpClient client = new OkHttpClient().newBuilder()  
  .build();  
MediaType mediaType = MediaType.parse("application/json");  
RequestBody body = RequestBody.create(mediaType, "");  
Request request = new Request.Builder()  
  .url("https://cas_server_url/oauth2.0/accessToken?grant_type=authorization_code&client_secret=xxx=&username=xxx&password=xxx&client_id=xxx")  
  .method("POST", body)  
  .addHeader("service", "https://my_server_url")  
  .addHeader("Content-Type", "application/json")  
  .build();  
Response response = client.newCall(request).execute();
```

access_token

:

- https://cas_server_url/oauth2.0/profile

https://cas_server_url/oauth2.0/profile?access_token=AT-3-rX4WfOiCgiF6qMNt52PfQDaQj-F4I9LK

- GET

					schema
X-App-Key	ID	header	true	string	

:

					schema
access_token	access_token	query	true	integer	

		schema
200	OK	

:

:

```
{
  "service" : "service_url",
  "attributes" : {
    "credentialType" : "LoginCredential"
  },
  "id" : "admin",
  "client_id" : "client"
}
```

:

```
{
  "code": 500,
  "data": {},
  "message": "",
  "status": "fail"
}
```

:

			schema
code		integer(int32)	
data		array	UserProfile
message		string	
status	,:success,fail,error	string	

Schema :

UserProfile

			schema
service		string	
attributes		integer(int64)	
id	ID	string	
client_id	ID	string	

:

JavaScript - Fetch

```
var myHeaders = new Headers();
myHeaders.append("Content-Type", "application/json");

var requestOptions = {
  method: 'POST',
  headers: myHeaders,
  redirect: 'follow'
};

fetch("https://cas_server_url/oauth2.0/profile?access_token=AT-8-dJTiv5R
IciPET1bJshNZHYs51m9FzJba", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

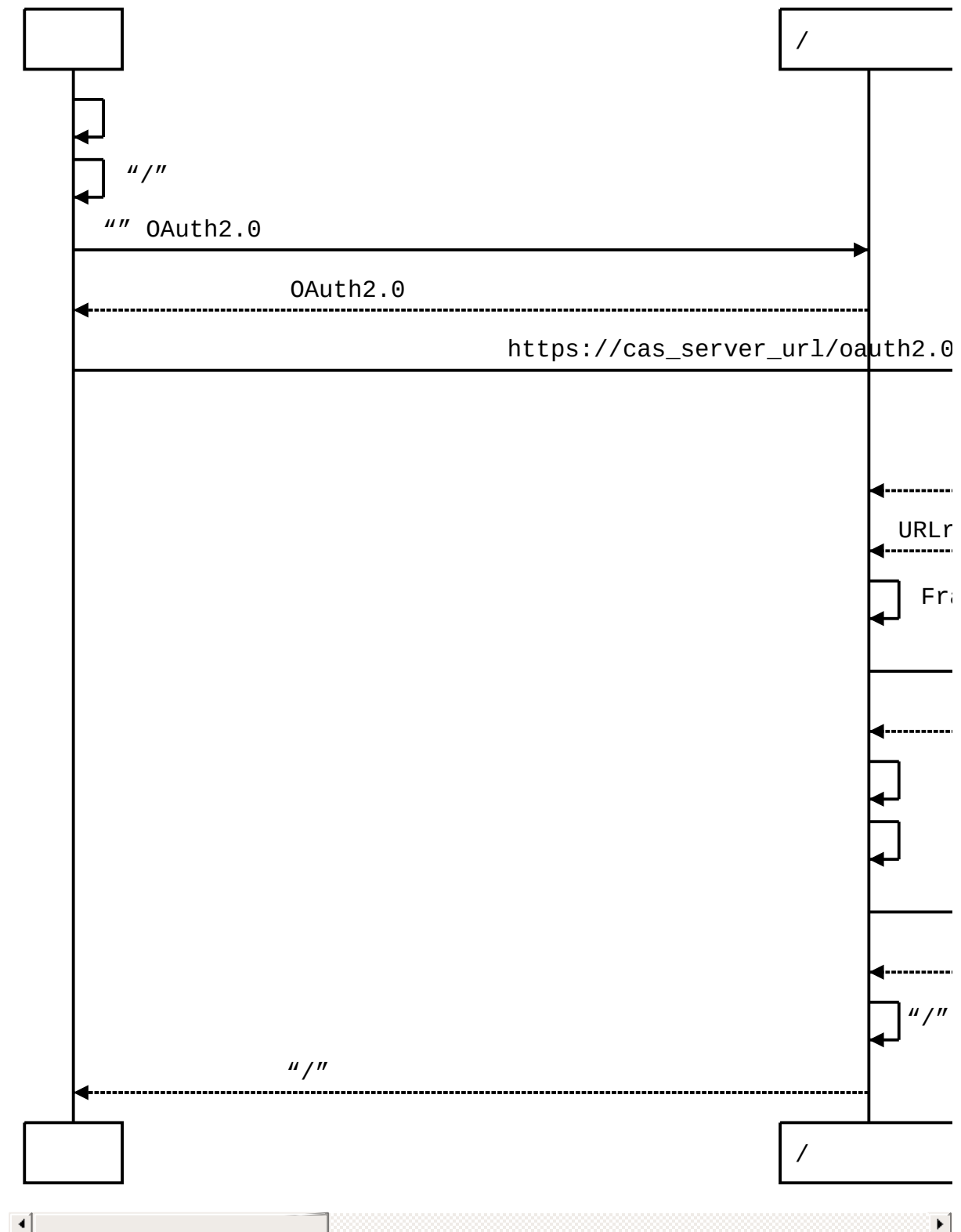
Java - OkHttp

```
OkHttpClient client = new OkHttpClient().newBuilder()
    .build();
MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "");
Request request = new Request.Builder()
    .url("https://cas_server_url/oauth2.0/profile?access_token=AT-8-dJTiv5
RiciPET1bJshNZHYs51m9FzJba")
    .method("POST", body)
    .addHeader("Content-Type", "application/json")
    .build();
Response response = client.newCall(request).execute();
```


3 Implicit

OAuth2.0 Implicit

JavaScript



access_token

JavaScript access_token https://cas_server_url/oauth2.0/authorize?response_type=token&client_id=yourclient&redirect_uri=https://my_server_url/callback

					schema
client_id	client_id	query	true	string	
response_type	token	query	true	string	
redirect_uri		query	true	string	

redirect_uri#

[https://my_server_url/callback#token=\\${access_token}](https://my_server_url/callback#token=${access_token})

access_token

```
URI uri = new UR("https://my_server_url/callback#token=access_token");
System.out.println(uri.getFragment());
```

access_token

accesstoken https://cas_server_url/oauth2.0/profile?access_token=xxxxx

```
{
  "service" : "service_url",
  "attributes" : {
    "credentialType" : "LoginCredential"
  },
  "id" : "admin",
  "client_id" : "client"
}
```

			schema
service		string	
attributes		integer(int64)	
id	ID	string	
client_id	ID	string	

access_token

access_token

access_token

:

- https://cas_server_url/oauth2.0/authorize

[https://cas_server_url/oauth2.0/authorize?
response_type=token&client_id=clientxxx&redirect_uri=https://my_server_url/callback](https://cas_server_url/oauth2.0/authorize?response_type=token&client_id=clientxxx&redirect_uri=https://my_server_url/callback)

- GET

					schema

:

					schema
client_id	client_id	query	true	string	
response_type	token	query	true	string	
redirect_uri		query	true	string	

redirect_uri#

[https://my_server_url/callback#token=\\${access_token}](https://my_server_url/callback#token=${access_token})

access_token

access_token

```
URIBuilder uribuilder = new URIBuilder("https://my_server_url/callback#token=access_token");
System.out.println(uribuilder.getFragment());
```

		schema
200	OK	

access_token

:

- https://cas_server_url/oauth2.0/profile

https://cas_server_url/oauth2.0/profile?access_token=AT-3-rX4WfOiCgiF6qMNt52PfQDaQj-F4I9LK

- GET

					schema
X-App-Key	ID	header	true	string	

:

					schema
access_token	access_token	query	true	integer	

		schema
200	OK	

:

:

```
{
  "service" : "service_url",
  "attributes" : {
    "credentialType" : "LoginCredential"
  },
  "id" : "admin",
  "client_id" : "client"
}
```

:

```
{
  "code": 500,
  "data": {},
  "message": "",
  "status": "fail"
}
```

:

			schema
code		integer(int32)	
data		array	UserProfile
message		string	
status	,:success,fail,error	string	

Schema :

UserProfile

			schema
service		string	
attributes		integer(int64)	
id	ID	string	
client_id	ID	string	

:

JavaScript - Fetch

```
var myHeaders = new Headers();
myHeaders.append("Content-Type", "application/json");

var requestOptions = {
  method: 'POST',
  headers: myHeaders,
  redirect: 'follow'
};

fetch("https://cas_server_url/oauth2.0/profile?access_token=AT-8-dJTiv5R
IciPET1bJshNZHYs51m9FzJba", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

Java - OkHttp

```
OkHttpClient client = new OkHttpClient().newBuilder()
    .build();
MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "");
Request request = new Request.Builder()
    .url("https://cas_server_url/oauth2.0/profile?access_token=AT-8-dJTiv5
RiciPET1bJshNZHYs51m9FzJba")
    .method("POST", body)
    .addHeader("Content-Type", "application/json")
    .build();
Response response = client.newCall(request).execute();
```

Spring MVC

Spring MVC

```
access_token , redirect_uri
```

1 access_token

```
https://cas_server_url/oauth2.0/authorize?
response_type=token&client_id=clientxxx&redirect_uri=https://my_server_url/callback
```

2 https://my_server_url/callback#token=access_token

```
/callback access_token
```

3 access_token

```
accesstoken https://cas_server_url/oauth2.0/profile?access_token=xxxxxx
```

:

```
{
  "service" : "service_url",
  "attributes" : {
    "credentialType" : "LoginCredential"
  },
  "id" : "admin",
  "client_id" : "client"
}
```

			schema
service		string	
attributes		integer(int64)	
id	ID	string	
client_id	ID	string	

```

import com.alibaba.fastjson2.JSONObject;
import lombok.extern.slf4j.Slf4j;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
import org.apache.http.NameValuePair;
import org.apache.http.client.utils.URLEncodedUtils;
import org.springframework.util.StringUtils;
import org.springframework.web.servlet.HandlerInterceptor;
import org.springframework.web.servlet.ModelAndView;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.net.URI;
import java.net.URISyntaxException;
import java.nio.charset.Charset;
import java.util.List;
import java.util.Optional;

@Slf4j
public class CasAuthInterceptor implements HandlerInterceptor {

    /**
     * HTTP Authorization Param, equal to <code>accessToken</code>
     */
    public static final String AUTHORIZATION_PARAM = "access_token";

    public OkHttpClient okHttpClient;

    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {
        System.out.println("preHandle.....");

        // 1.accessToken
        String accessToken = this.obtainAccessToken(request);
        if(StringUtils.hasText(accessToken)){

            // 2.Request.Builder
            long startTime = System.currentTimeMillis();
            Request.Builder builder = new Request.Builder().url("https://cas_server_url/oauth2.0/profile?access_token=" + accessToken);
            // 2.call, Request
            try {
                Response okResponse = okHttpClient.newCall(builder.get().build()).execute();
                if (okResponse.isSuccessful()) {

```

```

        log.info("OkHttp3 >> Request Success : code : {}, use time : {} ", okResponse.code(), System.currentTimeMillis() - startTime);
    }
    /**
     *
     * {
     *     "service" : "service_url",
     *     "attributes" : {
     *         "credentialType" : "LoginCredential"
     *     },
     *     "id" : "admin",
     *     "client_id" : "client"
     * }
     */
    String body = okResponse.body().string();
    JSONObject jsonObject = JSONObject.parseObject(body);
;
    // ID
    String clientId = jsonObject.getString("client_id");
    //
    JSONObject attributes = jsonObject.getJSONObject("attributes");
    } else {
        log.error("OkHttp3 >> Request Failure : code : {}, message : {}, use time : {} ", okResponse.code(), okResponse.message(), System.currentTimeMillis() - startTime);
    }
    } catch (Exception e) {
        log.error("OkHttp3 Request Error : {}, use time : {}", e.getMessage(), System.currentTimeMillis() - startTime);
    }
    return true;
}
response.sendRedirect("https://cas_server_url/oauth2.0/authorize?response_type=token&client_id=clientxxx&redirect_uri=https://my_server_url/callback");
return false;
}

@Override
public void postHandle(HttpServletRequest request, HttpServletResponse response, Object handler, ModelAndView modelAndView) throws Exception {
    System.out.println("postHandle.....");
}

@Override
public void afterCompletion(HttpServletRequest request, HttpServletResponse response, Object handler, Exception ex) throws Exception {

```

```

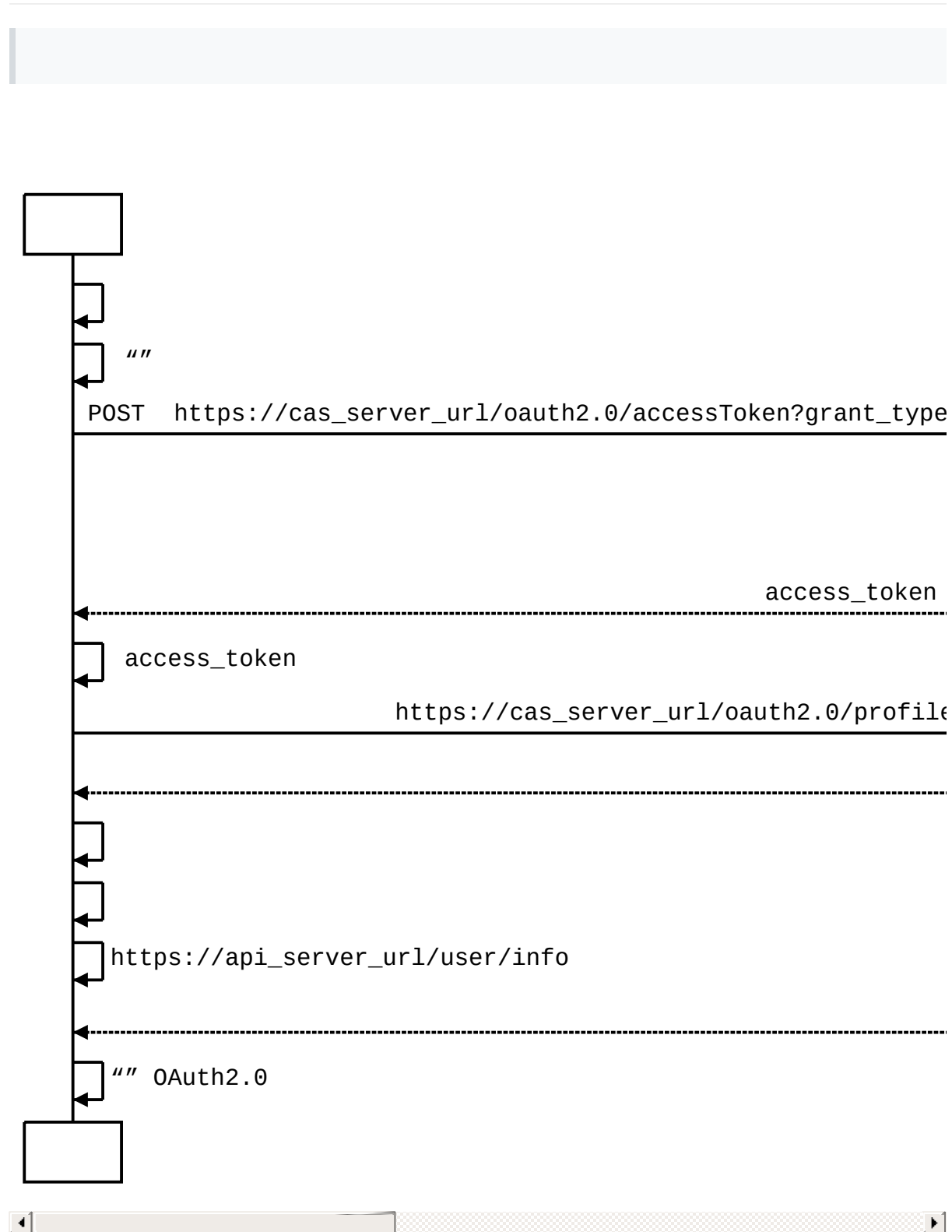
        System.out.println("afterCompletion.....");
    }

    protected String obtainAccessToken(HttpServletRequest request) throws URISyntaxException {
        // headerAcessToken
        URI uri = new URI(request.getRequestURL().toString());
        // https://my_server_url/callback#token=access_token
        System.out.println(uri.getFragment());
        List<NameValuePair> valuePairList = URLEncodedUtils.parse(uri.getFragment(), Charset.defaultCharset());
        Optional<NameValuePair> optional = valuePairList.stream().filter(
            nameValuePair -> nameValuePair.getName().equals("token")).findFirst();
        if(optional.isPresent()){
            return optional.get().getValue();
        }
        String accessToken = request.getParameter(AUTHORIZATION_PARAM);
        return accessToken;
    }
}

```

4Password Credentials

OAuth2.0 Password Credentials



Restfull API access_token

[https://cas_server_url/oauth2.0/accessToken?
grant_type=password&client_id=client&username=admin&password=admin](https://cas_server_url/oauth2.0/accessToken?grant_type=password&client_id=client&username=admin&password=admin)

					schema
grant_type	password	query	true	string	
client_id	client_id	query	true	string	
client_secret	client_secret	query	true	string	
username		query	true	string	
password		query	true	string	

```
{
  "access_token": "AT-3-rX4Wf0iCgiF6qMNt52PfQDaQj-F4I9LK",
  "token_type": "bearer",
  "expires_in": 28800
}
```

access_token

accesstoken https://cas_server_url/oauth2.0/profile?access_token=xxxxx

```
{
  "service" : "service_url",
  "attributes" : {
    "credentialType" : "LoginCredential"
  },
  "id" : "admin",
  "client_id" : "client"
}
```

			schema
service		string	

attributes		integer(int64)	
id	ID	string	
client_id	ID	string	

access_token

access_token

Restfull API access_token access_token

:

- https://cas_server_url/oauth2.0/accessToken

[https://cas_server_url/oauth2.0/accessToken?
grant_type=password&client_id=client&client_secret=123456&username=admin&password=ad
min](https://cas_server_url/oauth2.0/accessToken?grant_type=password&client_id=client&client_secret=123456&username=admin&password=admin)

- GET
- POST

:

					schema
grant_type	password	query	true	string	
client_id	client_id	query	true	string	
client_secret	client_secret	query	true	string	
username		query	true	string	
password		query	true	string	

		schema
200	OK	

:

access_token

:

```
{
  "access_token": "AT-3-rX4Wf0iCgiF6qMnt52PfQDaQj-F4I9LK",
  "token_type": "bearer",
  "expires_in": 28800
}
```

:

```
{
  "timestamp": 1660987198227,
  "status": 401,
  "error": "Unauthorized",
  "message": "No message available",
  "path": "/oauth2.0/accessToken"
}
```

:

			schema
access_token	access_token	string	
token_type	bearer	string	
expires_in		integer(int32)	

:

JavaScript - Fetch

```
var myHeaders = new Headers();
myHeaders.append("service", "https://my_server_url");
myHeaders.append("Content-Type", "application/json");

var requestOptions = {
  method: 'POST',
  headers: myHeaders,
  redirect: 'follow'
};

fetch("https://cas_server_url/oauth2.0/accessToken?grant_type=password&client_secret=xxx&username=xxx&password=xxx&client_id=xxx", requestOptions)
  .then(response => response.text())
```

access_token

```
.then(result => console.log(result))  
.catch(error => console.log('error', error));
```

Java - OkHttp

```
OkHttpClient client = new OkHttpClient().newBuilder()  
    .build();  
MediaType mediaType = MediaType.parse("application/json");  
RequestBody body = RequestBody.create(mediaType, "");  
Request request = new Request.Builder()  
    .url("https://cas_server_url/oauth2.0/accessToken?grant_type=password&  
client_secret=xxx=&username=xxx&password=xxx&client_id=xxx")  
    .method("POST", body)  
    .addHeader("service", "https://my_server_url")  
    .addHeader("Content-Type", "application/json")  
    .build();  
Response response = client.newCall(request).execute();
```

access_token

:

- https://cas_server_url/oauth2.0/profile

https://cas_server_url/oauth2.0/profile?access_token=AT-3-rX4WfOiCgiF6qMNt52PfQDaQj-F4I9LK

- GET

					schema
X-App-Key	ID	header	true	string	

:

					schema
access_token	access_token	query	true	integer	

		schema
200	OK	

:

:

```
{
  "service" : "service_url",
  "attributes" : {
    "credentialType" : "LoginCredential"
  },
  "id" : "admin",
  "client_id" : "client"
}
```

:

```
{
  "code": 500,
  "data": {},
  "message": "",
  "status": "fail"
}
```

:

			schema
code		integer(int32)	
data		array	UserProfile
message		string	
status	,:success,fail,error	string	

Schema :

UserProfile

			schema
service		string	
attributes		integer(int64)	
id	ID	string	
client_id	ID	string	

:

JavaScript - Fetch

```
var myHeaders = new Headers();
myHeaders.append("Content-Type", "application/json");

var requestOptions = {
  method: 'POST',
  headers: myHeaders,
  redirect: 'follow'
};

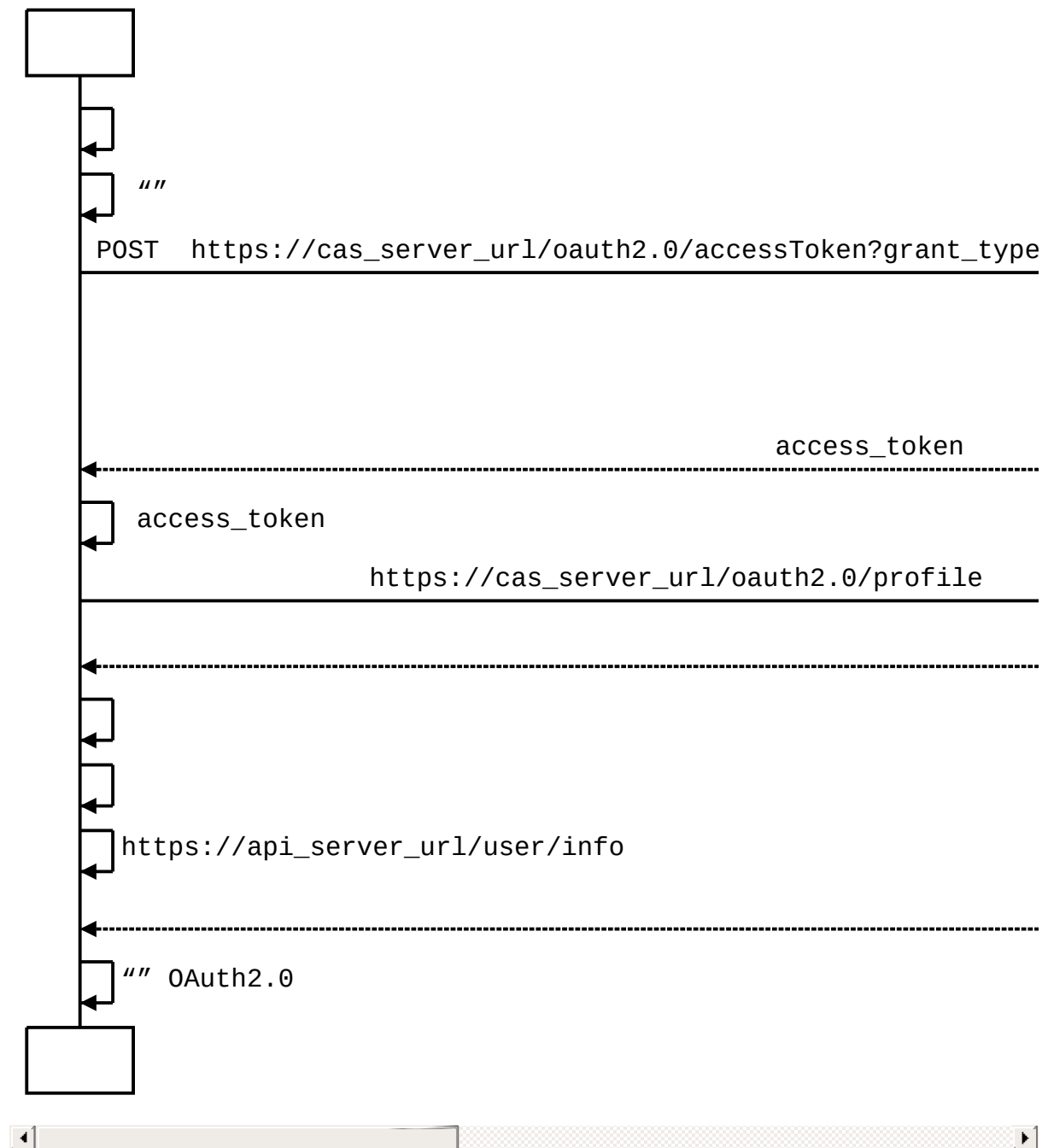
fetch("https://cas_server_url/oauth2.0/profile?access_token=AT-8-dJTiv5R
IciPET1bJshNZHYs51m9FzJba", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

Java - OkHttp

```
OkHttpClient client = new OkHttpClient().newBuilder()
    .build();
MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "");
Request request = new Request.Builder()
    .url("https://cas_server_url/oauth2.0/profile?access_token=AT-8-dJTiv5
RiciPET1bJshNZHYs51m9FzJba")
    .method("POST", body)
    .addHeader("Content-Type", "application/json")
    .build();
Response response = client.newCall(request).execute();
```

5Client Credentials

OAuth2.0 Client Credentials



Restfull API access_token

[https://cas_server_url/oauth2.0/accessToken?
grant_type=client_credentials&client_id=client&client_secret=123456](https://cas_server_url/oauth2.0/accessToken?grant_type=client_credentials&client_id=client&client_secret=123456)

					schema
grant_type	client_credentials	query	true	string	
client_id	client_id	query	true	string	
client_secret	client_secret	query	true	string	

```
{
  "access_token": "AT-3-rX4Wf0iCgiF6qMnt52PfQDaQj-F4I9LK",
  "token_type": "bearer",
  "expires_in": 28800
}
```

access_token

accesstoken https://cas_server_url/oauth2.0/profile?access_token=xxxxx

```
{
  "service" : "service_url",
  "attributes" : {
    "credentialType" : "LoginCredential"
  },
  "id" : "admin",
  "client_id" : "client"
}
```

			schema
service		string	
attributes		integer(int64)	
id	ID	string	

client_id	ID	string	
-----------	----	--------	--

access_token

access_token

Restfull API access_token access_token

:

- https://cas_server_url/oauth2.0/accessToken

[https://cas_server_url/oauth2.0/accessToken?
grant_type=client_credentials&client_id=client&client_secret=123456](https://cas_server_url/oauth2.0/accessToken?grant_type=client_credentials&client_id=client&client_secret=123456)

- GET
- POST

:

					schema
grant_type	client_credentials	query	true	string	
client_id	client_id	query	true	string	
client_secret	client_secret	query	true	string	

		schema
200	OK	

:

:

{

```

    "access_token": "AT-3-rX4Wf0iCgiF6qMnt52PfQDaQj-F4I9LK",
    "token_type": "bearer",
    "expires_in": 28800
  }

```

:

```

{
  "timestamp": 1660987198227,
  "status": 401,
  "error": "Unauthorized",
  "message": "No message available",
  "path": "/oauth2.0/accessToken"
}

```

:

			schema
access_token	access_token	string	
token_type	bearer	string	
expires_in		integer(int32)	

:

JavaScript - Fetch

```

var myHeaders = new Headers();
myHeaders.append("Content-Type", "application/json");

var requestOptions = {
  method: 'POST',
  headers: myHeaders,
  redirect: 'follow'
};

fetch("https://cas_server_url/oauth2.0/accessToken?grant_type=client_credentials&client_id=xxx&client_secret=xxx", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));

```

Java - OkHttp

```
OkHttpClient client = new OkHttpClient().newBuilder()
    .build();
MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "");
Request request = new Request.Builder()
    .url("https://cas_server_url/oauth2.0/accessToken?grant_type=client_credentials&client_id=xxx&client_secret=xxx")
    .method("POST", body)
    .addHeader("Content-Type", "application/json")
    .build();
Response response = client.newCall(request).execute();
```

access_token

:

- https://cas_server_url/oauth2.0/profile

https://cas_server_url/oauth2.0/profile?access_token=AT-3-rX4WfOiCgiF6qMNt52PfQDaQj-F4I9LK

- GET

					schema
X-App-Key	ID	header	true	string	

:

					schema
access_token	access_token	query	true	integer	

		schema
200	OK	

:

:

```
{
  "service" : "service_url",
  "attributes" : {
    "credentialType" : "LoginCredential"
  },
  "id" : "admin",
  "client_id" : "client"
}
```

:

```
{
  "code": 500,
  "data": {},
  "message": "",
  "status": "fail"
}
```

:

			schema
code		integer(int32)	
data		array	UserProfile
message		string	
status	,:success,fail,error	string	

Schema :

UserProfile

			schema
service		string	
attributes		integer(int64)	
id	ID	string	
client_id	ID	string	

:

JavaScript - Fetch

```
var myHeaders = new Headers();
myHeaders.append("Content-Type", "application/json");

var requestOptions = {
  method: 'POST',
  headers: myHeaders,
  redirect: 'follow'
};

fetch("https://cas_server_url/oauth2.0/profile?access_token=AT-8-dJTiv5R
IciPET1bJshNZHYs51m9FzJba", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

Java - OkHttp

```
OkHttpClient client = new OkHttpClient().newBuilder()
    .build();
MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "");
Request request = new Request.Builder()
    .url("https://cas_server_url/oauth2.0/profile?access_token=AT-8-dJTiv5
RiciPET1bJshNZHYs51m9FzJba")
    .method("POST", body)
    .addHeader("Content-Type", "application/json")
    .build();
Response response = client.newCall(request).execute();
```

6OAuth2.0

6.1Apache Oltu

Apache Oltu OAuth OAuth 1.0 OAuth 2.0 OAuth Web Oltu
Apache Java OAuth OAuth

Cas

- Spring MVC Web Cas
- Spring Boot Cas
- Spring Boot Cas

```
<!-- https://mvnrepository.com/artifact/org.apache.oltu.oauth2/org.apache.oltu.oauth2.client -->  
<dependency>  
  <groupId>org.apache.oltu.oauth2</groupId>  
  <artifactId>org.apache.oltu.oauth2.client</artifactId>  
  <version>1.0.2</version>  
</dependency>
```

6.1.1Spring MVC

6.2.2Spring Boot

Spring Boot OAuth 2.0

Apache Oltu OAuth OAuth 1.0 OAuth 2.0 OAuth Web

1

org.apache.oltu.oauth2.client

Maven

```
<properties>
  <oltu.oauth2.version>1.0.2</oltu.oauth2.version>
  <oltu.oauth2-jwt.version>1.0.3</oltu.oauth2-jwt.version>
</properties>

<!-- https://mvnrepository.com/artifact/org.apache.oltu.oauth2/org.apache.oltu.oauth2.client -->
<dependency>
  <groupId>org.apache.oltu.oauth2</groupId>
  <artifactId>org.apache.oltu.oauth2.client</artifactId>
  <version>1.0.2</version>
</dependency>
<dependency>
  <groupId>org.apache.oltu.oauth2</groupId>
  <artifactId>org.apache.oltu.oauth2.dynamicreg.client</artifactId>
</dependency>
```

Gradle

```
// https://mvnrepository.com/artifact/org.apache.oltu.oauth2/org.apache.oltu.oauth2.client
implementation group: 'org.apache.oltu.oauth2', name: 'org.apache.oltu.oauth2.client', version: '1.0.2'
```

2

GetAuthorizationController

```
import lombok.extern.slf4j.Slf4j;
import org.apache.oltu.oauth2.client.OAuthClient;
import org.apache.oltu.oauth2.client.URLConnectionClient;
```

```

import org.apache.oltu.oauth2.client.request.OAuthBearerClientRequest;
import org.apache.oltu.oauth2.client.request.OAuthClientRequest;
import org.apache.oltu.oauth2.client.response.OAuthResourceResponse;
import org.apache.oltu.oauth2.common.OAuth;
import org.apache.oltu.oauth2.common.exception.OAuthProblemException;
import org.apache.oltu.oauth2.common.exception.OAuthSystemException;
import org.apache.oltu.oauth2.common.message.types.GrantType;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

import javax.servlet.http.HttpServletRequest;

/**
 *
 * https://blog.csdn.net/winerpro/article/details/133266414
 */
@Controller
@RequestMapping("/client")
@Slf4j
public class GetAuthorizationController {

    private static String CLIENT_ID = "20210903";
    private static String CLIENT_SECRET = "cas123456";
    private static String CODE_URL = "oauth2.0/authorize";
    private static String RESPONSE_TYPE = "code";
    /**/
    private static String AUTH_SERVER_URL = "https://cas.test.com:8443/cas/oauth2.0/accessToken";
    /**/
    private static String RESOURCE_OWNER_URL = "https://cas.test.com:8443/cas/";
    /**/
    private static String RESOURCE_SERVER_URL = "https://cas.test.com:8443/cas/oauth2.0/profile";
    /**/
    private static String CALLBACKCODE = "https://localhost:8082/client/callbackCode";
    /**/
    private static String GETRESOURCE = "https://localhost:8082/client/getResource";

    /**
     *
     */
    @GetMapping("/getCode")
    public String getCode() throws OAuthSystemException {

```

```

        OAuthClientRequest oAuthClientRequest = OAuthClientRequest
            .authorizationLocation(CODE_URL)
            .setClientId(CLIENT_ID)
            .setResponseType(RESPONSE_TYPE)
            .setRedirectURI(CALLBACKCODE)
            .buildQueryMessage();
        String uriString = oAuthClientRequest.getLocationUri();
        //
        return "redirect:" + RESOURCE_OWNER_URL + uriString;
    }

    /**
     * 1.
     * 2.
     */
    @GetMapping("/callbackCode")
    public String callbackCode(HttpServletRequest request) throws OAuthSystemException, OAuthProblemException {
        String code = request.getParameter("code");
        log.info(" --- code: {} -----", code);
        //code
        OAuthClient oAuthClient = new OAuthClient(new URLConnectionClient());
        OAuthClientRequest tokenRequest = OAuthClientRequest
            .tokenLocation(AUTH_SERVER_URL)
            .setClientId(CLIENT_ID)
            .setClientSecret(CLIENT_SECRET)
            .setGrantType(GrantType.AUTHORIZATION_CODE)
            .setCode(code)
            .setRedirectURI(GETRESOURCE)
            .buildQueryMessage();
        //Code
        OAuthResourceResponse resourceResponse = oAuthClient.resource(tokenRequest, OAuth.HttpMethod.GET, OAuthResourceResponse.class);
        String accessToken = resourceResponse.getBody();
        //
        log.info(" " + accessToken);
        return "redirect:" + GETRESOURCE + "?" + accessToken;
    }

    /**
     *
     */
    @GetMapping("/getResource")
    @ResponseBody
    public String getResource(String access_token) throws OAuthSystemException, OAuthProblemException {
        log.info(" token = " + access_token);
    }

```

```

        OAuthClient oAuthClient = new OAuthClient(new URLConnectionClien
t());
        OAuthClientRequest userInfoRequest = new OAuthBearerClientReques
t(RESOURCE_SERVER_URL)
            .setAccessToken(access_token)
            .buildQueryMessage();
        OAuthResourceResponse resourceResponse = oAuthClient.resource(us
erInfoRequest, OAuth.HttpMethod.GET, OAuthResourceResponse.class);
        String userInfo = resourceResponse.getBody();
        log.info(" " + userInfo);
        return userInfo;
    }
}

```

3

<https://localhost:8082/client/getCode>

,CAS

CAS

Login

请输入您的用户名和密码.

用户名:

密 码:

登录

[Forgot your password?](#)

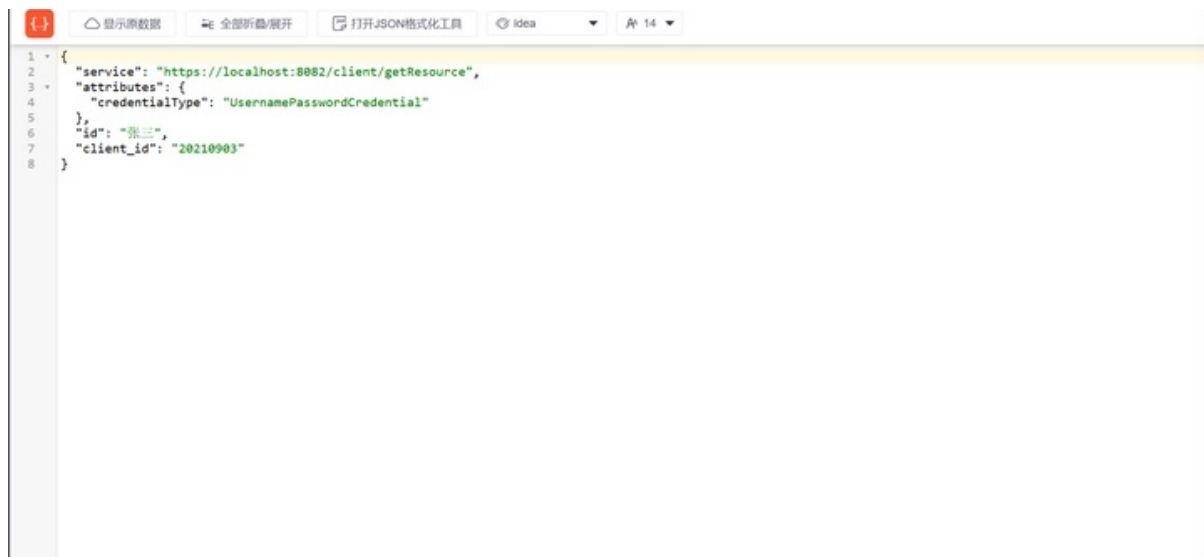
出于安全考虑，一旦您访问过那些需要提供凭证信息的应用时，请操作完成之后[登出](#)并关闭浏览器。

OAuthService

Links to CAS Resources

- [Dashboard](#)
- [Documentation](#)
- [Pull Requests](#)
- [Contributor Guidelines](#)
- [Mailing Lists](#)
- [Chatroom](#)
- [Blog](#)

11



The screenshot shows an IDE window with a code editor displaying a JSON object. The JSON object is as follows:

```
1 {  
2   "service": "https://localhost:8082/client/getResource",  
3   "attributes": {  
4     "credentialType": "UsernamePasswordCredential"  
5   },  
6   "id": "张三",  
7   "client_id": "20210903"  
8 }
```

The IDE interface includes a toolbar at the top with buttons for "显示源数据" (Show Source Data), "全部折叠/展开" (Collapse/Expand All), and "打开JSON格式化工具" (Open JSON Formatting Tool). The editor is set to "Idea" theme and font size "14".

Spring OAuth2 Client

```
<!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-  
boot-starter-oauth2-client -->  
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-oauth2-client</artifactId>  
  <version>2.3.12.RELEASE</version>  
</dependency>
```


6.1

WS Federation

SAML(v1v2)

1

SAML2.0 service providerSP XML

identity provider(IDP) Metadata XML

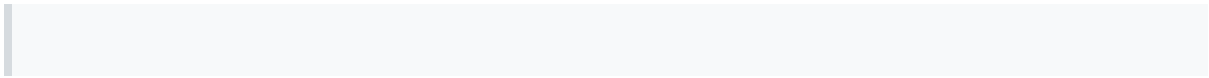
SPShibbolethIDP

:IDP XMLIDP Metadata

:SP shibboleth2.xml

,SP

```
<MetadataProvider type="XML" file="IDP.xml"/>
```



:

- https://cas_server_url/logout
- GET

					schema

:

					schema

:

OAuth 2.0

1Cas

:

- https://cas_server_url/logout
- GET

					schema

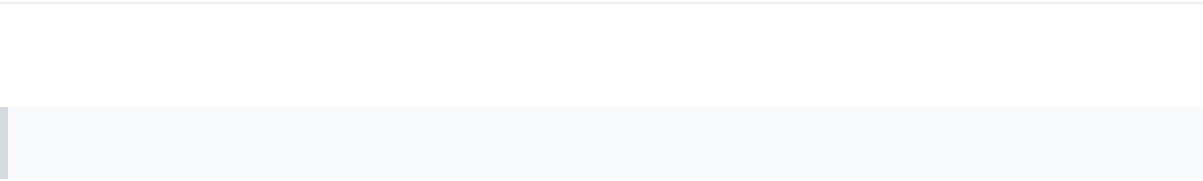
:

					schema

:

OAuth 2.0

2OAuth2.0



:

- https://cas_server_url/logout
- GET

					schema

:

					schema

:

OAuth 2.0

Cas

Markdown

Markdown

Markdown

Markdown Markdown John Gruber 2004

-
-
-

Markdown Word Word Markdown Markdown
Markdown

Heading One **this text is bold** Markdown
Markdown Visual Studio Code

Markdown 是什么？

Markdown 是一种轻量级的标记语言，可用于在纯文本文档中添加格式化元素。Markdown 由 John Gruber 于 2004 年创建，如今已成为世界上最受欢迎的标记语言之一。

1. 专注于文字内容；

2. 纯文本，易读易写，可以方便地纳入版本控制；

3. 语法简单，没有什么学习成本，能轻松在码字的同时做出美观大方的排版。

使用 Markdown 与使用 Word 类编辑器不同。在 Word 之类的应用程序中，单击按钮以设置单词和短语的格式，并且，更改立即可见。而 Markdown 与此不同，当你创建 Markdown 格式的文件时，可以在文本中添加 Markdown 语法，以指示哪些单词和短语看起来应该有所不同。

例如，要表示标题，只须在短语前面添加一个井号即可（例如，`# Heading One`）。或者要加粗一个短语，只须在短语前后各加两个星号即可（例如，`**this text is bold**`）。可能需要一段时间才能习惯在文本中看到 Markdown 语法，尤其是如果你已习惯了所见即所得的应用程序。

Markdown macOSWindowsLinuxiOS Android Markdown Web
Markdown

Gruber Markdown Markdown

Markdown Markdown

Markdown

Markdown Markdown word

- Markdown StackOverflowCSDNGitBookV2EX GitHubGitLab
BitBucketCodingGitee Markdown READMEWiki Markdown
- Markdown Markdown Markdown Markdown Markdown
Microsoft Word Microsoft Word

- Markdown Markdown
- Markdown Markdown

Markdown Markdown Markdown Markdown

Markdown Markdown

Markdown Markdown Markdown Markdown

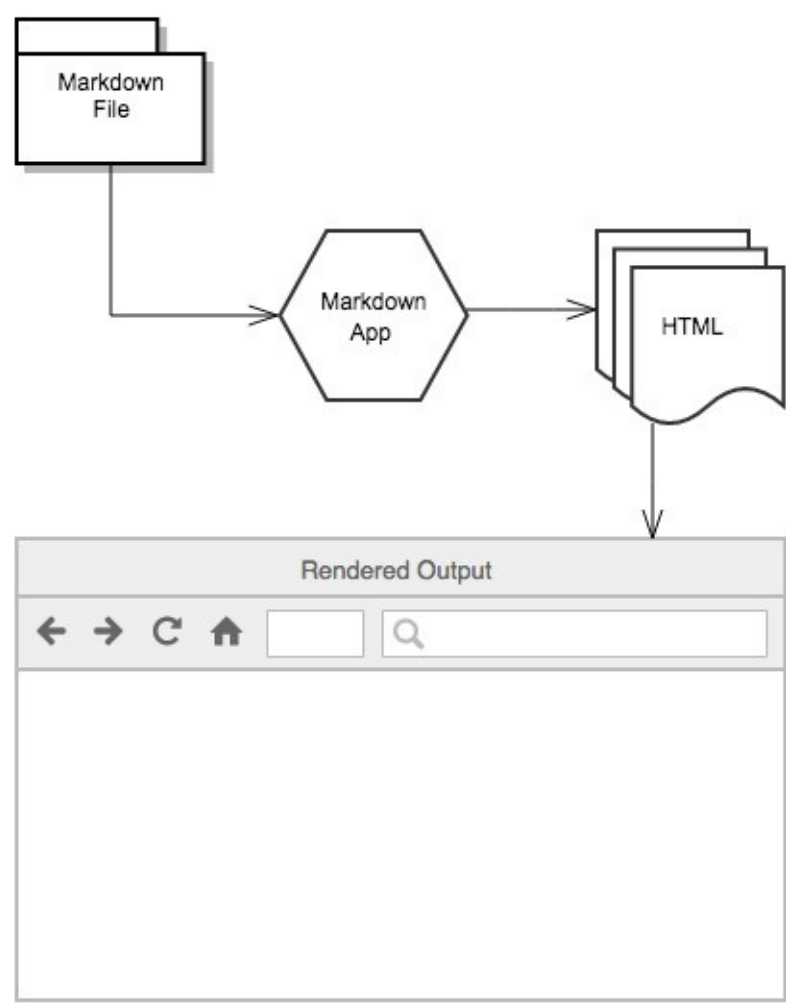
Markdown

Markdown .md .markdown Markdown HTML

Markdown Markdown Microsoft Word Markdown
Markdown HTML Web

Markdown Markdown “” Markdown HTML Web

Markdown “Markdown ”



- 1. Markdown Markdown .md .markdown
- 2. Markdown Markdown
- 3. Markdown Markdown HTML
- 4. web HTML Markdown PDF

Markdown 1-3

Markdown

Markdown

Markdown Markdown Markdown Markdown

Markdown

Markdown web

- [Hexo](#)
- [Hugo](#)
- [Jekyll](#)
- [Gatsby](#)
- [VuePress](#)
- [Docusaurus](#)

Markdown Microsoft Word Markdown Markdown PDF HTML
PDF PDF

Markdown

- VSCode / Atom
- Vim / Emacs / Sublime Text / Notepad++
- IDE
IntelliJ IDEA / Android Studio / WebStorm
- Ulysses / Mou / Typora / Markpad
- Markdown Markdown

Markdown Markdown

- Notion
-
-
-
-

Markdown Markdown
RemarkGitHub CleaverGitHub Markdown Mac Deckset

PowerPoint

Keynote

Marked

Markdown [Markdown Here] Markdown HTML

Markdown GitHub Markdown

- Read the Docs Markdown GitHub Read the Docs
- MkDocs Markdown YAML MkDocs Read the Docs
MkDocs Material
- Docusaurus
- VuePress Vue

Markdown

Markdown Markdown Markdown Markdown flavors Markdown

Markdown language dialects Markdown Dillinger Markdown
Ulysses

“Markdown”

Markdown Markdown Markdown Markdown

https://blog.csdn.net/qq_63585949/article/details/131435580

Markdown

Markdown

Markdown

HTML

HTMLPDF

Markdown

GithubWikipedia

.md

Markdown

Markdown

(

#

)

#

#

(

<h3>

)(

My Header

)

Markdown	HTML	
<div># Heading level 1</div>	<div><h1>Heading level 1</h1></div>	<div>Heading level 1</div>
<div>## Heading level 2</div>	<div><h2>Heading level 2</h2></div>	<div>Heading level 2</div>
<div>### Heading level 3</div>	<div><h3>Heading level 3</h3></div>	<div>Heading level 3</div>
<div>#### Heading level 4</div>	<div><h4>Heading level 4</h4></div>	<div>Heading level 4</div>
<div>##### Heading level 5</div>	<div><h5>Heading level 5</h5></div>	<div>Heading level 5</div>
<div>##### Heading level 6</div>	<div><h6>Heading level 6</h6></div>	<div>Heading level 6</div>

==

--

Markdown	HTML	
<div>Heading level 1</div> <div>=====</div>	<div><h1>Heading level</div> <div>1</h1></div>	<div>=====</div> <div>Heading level 1</div> <div>_____</div>
<div>Heading level 2 -----</div> <div>-----</div>	<div><h2>Heading level</div> <div>2</h2></div>	<div>_____</div> <div>Heading level 2</div> <div>_____</div>

Markdown

#

#

Do this Don't do this

Here's a Heading #Here's a Heading

Markdown

Markdown HTML

I really like using Markdown.

I think I'll use it to format all of my documents from now on.

I really like using Markdown.

I think I'll use it to format all of my documents from now on.

I really like using Markdown.

I think I'll use it to format all of my documents from now on.

spacestabs

Do this Don't do this
Don't put tabs or spaces in front of your paragraphs.

Keep lines left-aligned like this. This can result in unexpected formatting problems.

Don't add tabs or spaces in front of paragraphs.

Markdown

```
.(  
)
```

Markdown HTML

This is the first line.

And this is the second line.

This is the first line.

And this is the second line.

This is the first line.

And this is the second line.

Markdown (trailing whitespace) Markdown HTML

"" HTML

CommonMark () Markdown Enter

Do this Don't do this

First line with two spaces after.

And the next line.

First line with the HTML tag after.

And the next line. First line with a backslash after.

And the next line.

First line with nothing after.

And the next line.

Markdown

Bold

asterisksunderscoresasterisks

Markdown HTML

I just love **bold text**. I just love **bold text**. I just love bold text.

I just love **bold text**. I just love **bold text**. I just love bold text.

Love**is**bold Love**is**bold Love**is**bold
Bold

Markdown asterisks

Do this Don't do this

Love**is**bold Love__is__bold

Italic

asteriskunderscore

Markdown HTML

Italicized text is the *cat's meow*. Italicized text is the *cat's meow*. Italicized text is the *cat's meow*.

Italicized text is the *cat's meow*. Italicized text is the *cat's meow*. Italicized text is the *cat's meow*.

Acatmeow Acatmeow Acatmeow

Italic

Do this Don't do this

Acatmeow A_cat_meow

Italic

Markdown HTML

This text is ***really important***. This text is ***really important***. This text is really important.

This text is ***really important***. This text is ***really important***. This text is really important.

This text is ***really important***. This text is ***really important***. This text is really important.

This text is ***really important***. This text is ***really important***. This text is really important.

This is really***very***important text. This is really***very***important text. This is really***very***important text.

Italic

Markdown

Do this Don't do this

This is really***very***important text. This is really__very__important text.

Markdown

>

Dorothy followed her through many of the beautiful rooms in her castle.

1

Dorothy followed her through many **of** the beautiful rooms **in** her castle.

>

Dorothy followed her through many of the beautiful rooms in her castle.

The Witch bade her clean the pots and kettles and sweep the floor and keep the fire fed with wood.

1

2

3

Dorothy followed her through many **of** the beautiful rooms **in** her castle.

The Witch bade her clean the pots **and** kettles **and** sweep the floor **and** keep the fire fed with wood.

>>

Dorothy followed her through many of the beautiful rooms in her castle.

The Witch bade her clean the pots and kettles and sweep the floor and keep the fire fed with wood.

1

2

3

Dorothy followed her through many **of** the beautiful rooms **in** her castle.

The Witch bade her clean the pots **and** kettles **and** sweep the floor **and** keep the fire fed with wood.

Markdown

class="markdown-heading">>The quarterly results look great!

- Revenue was off the chart.
- Profits were higher than ever.

Everything is going according to **plan**.

1

2

3

4

5

6

The quarterly results look great

Revenue was **off** the chart.

Profits were higher than ever.

Everything **is** going according to plan.

Markdown

1

Markdown HTML

1. First item
2. Second item
3. Third item
4. Fourth item
 - i. First item
 - ii. Second item
 - iii. Third item
 - iv. Fourth item

First item

Second item

Third item

Fourth item

5. First item
6. Second item
7. Third item
8. Fourth item
 - i. First item
 - ii. Second item
 - iii. Third item
 - iv. Fourth item

First item
Second item
Third item
Fourth item

9. First item
10. Second item
11. Third item
12. Fourth item
 - i. First item
 - ii. Second item
 - iii. Third item
 - iv. Fourth item

First item
Second item
Third item
Fourth item

13. First item
14. Second item
15. Third item
 - i. Indented item
 - ii. Indented item
16. Fourth item
 - i. First item

- ii. Second item
- iii. Third item
 - a. Indented item
 - b. Indented item
- iv. Fourth item

First item
Second item
Third item

Indented item
Indented item

Fourth item

CommonMark () Markdown (.)

Do this Don't do this

1. First item
2. Second item
3. Third item
4. Fourth item
 - 1) First item
 - 2) Second item
 - 3) Third item
 - 4) Fourth item

(-) (*) (+)

Markdown HTML

- First item
- Second item
- Third item
- Fourth item
 - First item
 - Second item
 - Third item
 - Fourth item

First item

Second item

Third item

Fourth item

- First item
- Second item
- Third item
- Fourth item
 - First item
 - Second item
 - Third item
 - Fourth item

First item

Second item

Third item

Fourth item

- First item
- Second item
- Third item
- Fourth item
 - First item
 - Second item
 - Third item
 - Fourth item

First item

Second item

Third item

Fourth item

- First item
- Second item
- Third item
 - Indented item
 - Indented item
- Fourth item
 - First item
 - Second item
 - Third item
 - Indented item
 - Indented item
 - Fourth item

First item

Second item

Third item

Indented item
Indented item

Fourth item

Markdown

Do this Don't do this

- First item
- Second item
- Third item
- Fourth item + First item
- Second item

- Third item
- Fourth item

- This is the first list item.
- Here's the second list item.

I need to add another paragraph below the second list item.

- And here's the third list item.

1
2
3
4
5
6

```
This is the first list item.
```

```
Here's the second list item.
```

```
I need to add another paragraph below the second list item.
```

```
And here's the third list item.
```

- This is the first list item.
- Here's the second list item.

```
A blockquote would look great below the second list item.
```

- And here's the third list item.

1
2
3
4
5
6

This is the first **list** item.

Here's the second **list** item.

A blockquote would look great below the second **list** item.

And here's the third **list** item.

1. Open the file.
2. Find the following code block on line 21:

```
<html>
  <head>
    <title>Test</title>
  </head>
```

3. Update the title to match the name of your website.

1
2
3
4
5
6
7
8
9

Open the file.

Find the following code block on line 21:

```
<html>
  <head>
    <title>Test</title>
  </head>
  1
  2
  3
  4
```

Update the title to match the name of your website.

1. Open the file containing the Linux mascot.
2. Marvel at its beauty.

3. Close the file.

```
1
2
3
4
5
6
```

Open the file containing the Linux mascot.

Marvel at its beauty.

Tux, the Linux mascot

Close the file.

1. First item

2. Second item

3. Third item

- Indented item
- Indented item

4. Fourth item

- 1
- 2
- 3
- 4
- 5
- 6

First item

Second item

Third item

Indented item

Indented item

Fourth item

Markdown

(☐) Markdown ☐ HTML At the command prompt, type . At the command prompt, type . At the command prompt, type nano.

(☐) Markdown ☐ HTML ☐ Use in your Markdown file. ☐ Use in your Markdown file. ☐ Use in your Markdown file.

<html>

<head>

</head>

</html>

1

2

3

4

```
1  
2  
3  
4
```

Note: fenced code blocks
Markdown

(***)** (—) (□)

```
1  
2  
3  
4  
5
```

Horizontal Rule

Do this Don't do this
Try to put a blank line before...

...and after a horizontal rule. Without blank lines, this would be a heading.

Don't do this!
Markdown

title

Markdown

HTML

Title

title title

```
1
```

Email

URL email

<https://blog.marquis.eu.org>
fake@example.com

```
1
2
```

<https://blog.marquis.eu.org>
fake@example.com

,

**

code

```
1
2
3
```

code

URL Markdown

```
[hobbit-hole][1]
[hobbit-hole] [1]
```

```
[label]:
URL
```

```
[1]: https://en.wikipedia.org/wiki/Hobbit#Lifestyle
[1]: https://en.wikipedia.org/wiki/Hobbit#Lifestyle "Hobbit lifestyles"
[1]: https://en.wikipedia.org/wiki/Hobbit#Lifestyle 'Hobbit lifestyles'
[1]: https://en.wikipedia.org/wiki/Hobbit#Lifestyle (Hobbit lifestyles)
[1]: <https://en.wikipedia.org/wiki/Hobbit#Lifestyle> "Hobbit lifestyles"
[1]: <https://en.wikipedia.org/wiki/Hobbit#Lifestyle> 'Hobbit lifestyles'
[1]: <https://en.wikipedia.org/wiki/Hobbit#Lifestyle> (Hobbit lifestyles)
```

Markdown

Markdown URL %20
Do this Don't do this
link [link](https://www.example.com/my great page)
Markdown

(!),

Markdown

HTML

1

Markdown

1

Markdown

Markdown \

* Without the backslash, this would be a bullet in an unordered list.

1

- Without the backslash, this would be a bullet in an unordered list.

\

,

•

—

}

[]

()

•

•

.

!

|

HTML < & < & HTML < &

& AT&T AT&T &

<http://images.google.com/images?num=30&q=larry+bird>

1

<http://images.google.com/images?num=30&q=larry+bird>

1

href HTML

Markdown & HTML &

©

1

Markdown

AT&T

1

Markdown

AT&T

1

< Markdown HTML < HTML Markdown

4 < 5

1

Markdown

4 < 5

1

Markdown < & HTML Markdown HTML HTML < &

HTML

Markdown HTML

Markdown HTML HTML HTML Markdown HTML Markdown

HTML ~~Markdown~~ ~~Markdown~~ ~~HTML~~ ~~HTML~~

~~Markdown~~ ~~HTML~~

HTML Markdown

This **word** is bold. This *word* is italic.

1

:

This word is bold. This word is italic.

tab Markdown

Markdown HTML

This **is** a regular paragraph.

Foo

This is another regular paragraph.

1
2
3
4
5
6
7
8
9

Markdown HTML HTML Markdown
HTML

Markdown Markdown HTML Markdown HTML

HTML

blank linestabsspaces HTML

HTML Markdown

italic **and bold**

CC

4.0 BY-SA

https://blog.csdn.net/qq_63585949/article/details/131442576

Markdown

Markdown

Markdown

John Gruber

URL Markdown Markdown
Markdown

Markdown Markdown

Markdown Gruber URL Markdown

CommonMark
GitHub Flavored Markdown (GFM)
Markdown Extra
MultiMarkdown
R Markdown

Markdown

Markdown
Markdown

—|

Syntax	Description
Header	Title
Paragraph	Text

1
2
3
4

Syntax Description
Header Title

Paragraph Text

Syntax	Description
Header	Title
Paragraph	Text

```
1
2
3
4
```

Tip: [Markdown Tables Generator](#) [Markdown](#)

:

Syntax	Description	Test Text
Header	Title	Here's this
Paragraph	Text	And more

```
1
2
3
4
```

Syntax Description Test Text
Header Title Here's this
Paragraph Text And more

,

HTML

HTML||
Markdown

Markdown Markdown



```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25
}
```

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25
}
```

Tip:

Markdown

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25
}
```

```
1
2
3
4
5
6
7
```

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 25
}
```

```
1
2
```

3

4

5

Markdown

[^1]-

[^1]: My footnote.

Here's a simple footnote,[^1] and here's a longer one.[^bignote]

[^1]: This is the first footnote.

[^bignote]: Here's one with multiple paragraphs and code.

Indent paragraphs to include them **in** the footnote.

```
`{ my code }`
```

Add **as** many paragraphs **as** you like.

1

2

3

4

5

6

7

8

9

10

11

Here's a simple footnote,¹ and here's a longer one.²

Indent paragraphs to include them in the footnote.

```
{ my code }
```

Add as many paragraphs as you like.

1

2

3

4

5

Markdown {#heading-ids}

Markdown ID | Markdown ID CSS ID ID

My Great Heading {#custom-id}

1

HTML

My Great Heading

1

ID (#headid)

ID ID

Markdown HTML

[Heading IDs](#) [Heading IDs](#) [Heading IDs](#)

IDURLHeading IDs

Markdown

Markdown

First Term

: This is the definition of the first term.

Second Term

: This is one definition of the second term.

: This is another definition of the second term.

1

2

3

4

5

6

HTML

First Term

This is the definition of the first term.

Second Term

This is one definition of the second term.

This is another definition of the second term.

1

2

3

4

5

6

7

First Term

This is the definition of the first term. Second Term

This is one definition of the second term.

This is another definition of the second term.

Markdown

~~

—

1

Markdown

Markdown - [][] [x] x

- Write the press release
- Update the website
- Contact the media

- 1
- 2
- 3

Write the press **release**

Update the website

Contact the media

Markdown Emoji

Markdown Markdown emoji shortcodes

Emojipedia Markdown Markdown Markdown HTML PDF

Tip: HTML UTF-8

Markdown

:tent:

:joy:

1

2

3



Note: Markdown

Markdown URL

<http://www.example.com>

Markdown

<http://www.example.com>

URL

URL URL

`http://www.example.com`

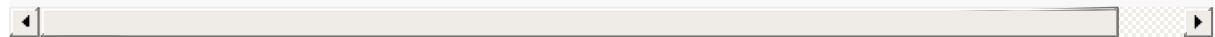
1

<http://www.example.com>

This **is** the first footnote. ↩

Here's one **with** multiple paragraphs **and** code. ↩

CC 4.0 BY-SA



https://blog.csdn.net/qq_63585949/article/details/131459826

Markdown Mermaid ()

Markdown Mermaid () - (Flowchart)

Mermaid

Mermaid JavaScript Markdown

Mermaid

- (Flowchart) : `graph` `flowchart`
- (Sequence Diagram) : `sequenceDiagram`
- (Class Diagram) : `classDiagram`
- (State Diagram) : `stateDiagram`
- (Entity Relationship Diagram) : `erDiagram`
- (User Journey Diagram) : `journey`
- (Gantt Diagram) : `gantt`
- (Pie Chart Diagram) : `pie`
- (Quadrant Chart) : `quadrantChart`
- (Requirement Diagram) : `requirementDiagram`
- Gitgraph (Gitgraph Diagram) : `gitGraph`
- (Mindmap) : `mindmap`
- (Timeline Diagram) : `timeline`

Mermaid

Mermaid

Mermaid

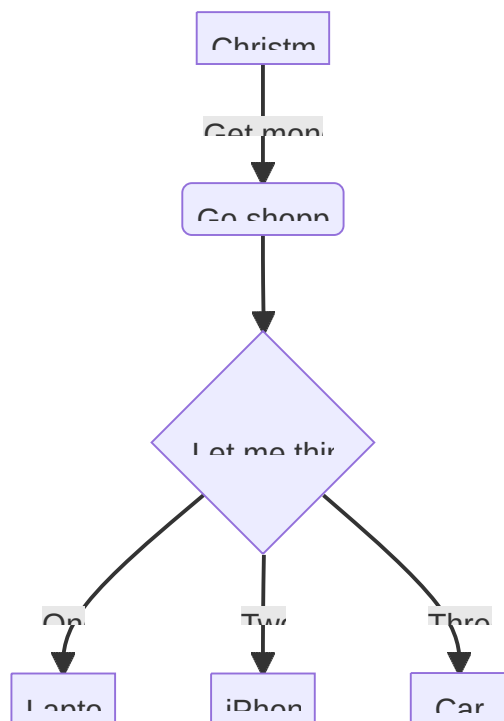
- Mermaid [Mermaid Live Editor](#)
- Mermaid Markdown [Typora](#)

Mermaid Mermaid

```
flowchart TD
    A[Christmas] -->|Get money| B(Go shopping)
```

```
B --> C{Let me think}  
C -->|One| D[Laptop]  
C -->|Two| E[iPhone]  
C -->|Three| F[fa:fa-car Car]
```

Mermaid Markdown Mermaid



Mermaid

(Flowchart)

Mermaid

```
flowchart LR
    id
```

```
graph TD
    flowchart
```




```
flowchart
```

```
id[]
```



Unicode

“” Unicode

```
flowchart
    id["❤️ Unicode"]
```



- TB (Top to bottom) :
- TD (Top-down) : TB
- BT (Bottom to top) :
- RL (Right to left) :
- LR (Left to right) :

```
flowchart TD
```

```
-->
```

```
flowchart LR
```

```
-->
```






```
flowchart LR
```

```
id1()
```



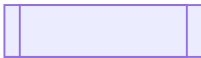
```
flowchart LR
```

```
id1([ ])
    
```



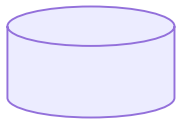
```
flowchart LR
```

```
id1[[] ]
```



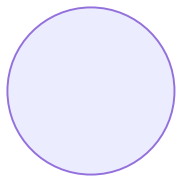
```
flowchart LR
```

```
id1[()]
```

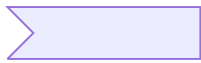



```
flowchart LR
```

```
id1(()
```

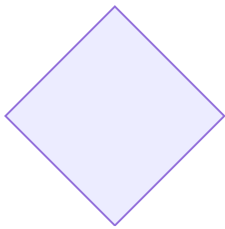


```
flowchart LR
    id1>]
```



```
flowchart LR
```

```
id1{}
```



```
flowchart LR
```

```
id1{{}}
```




```
flowchart LR
    id1[// ]
```

```
flowchart LR
    id1[\\]
```



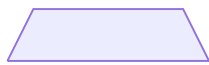


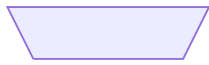
```
flowchart LR
```

```
id1[/\]
```

```
flowchart LR
```

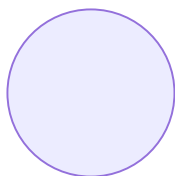
```
id1[\/]
```





```
flowchart LR
```

```
id1(()
```




```
flowchart LR
```

```
A-->B
```



```
flowchart LR
```

```
A---B
```



```
flowchart LR
```

```
A-- ---B
```

```
flowchart LR
```

```
A---||B
```



```
flowchart LR
```

```
A -- --> B
```

```
flowchart LR
```

```
A --> || B
```





```
flowchart LR
```

```
A-.->B
```



```
flowchart LR
```

```
A-. .-> B
```



```
flowchart LR
```

```
  A ==> B
```



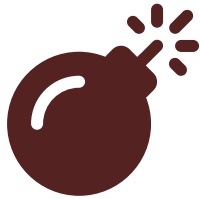
```
flowchart LR
```

```
  A ==  =  => B
```




```
flowchart LR
```

```
A ~~~ B
```



Syntax error in graph
mermaid version 8.13.0

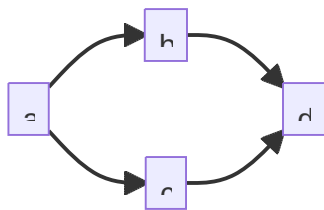
```
flowchart LR
```

```
A -- 1 --> B -- 2 --> C
```



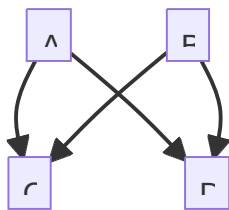
```
flowchart LR
```

```
  a --> b & c --> d
```



```
flowchart TB
```

```
A & B--> C & D
```



4

```
flowchart TB
```

```
A --> C
```

```
A --> D
```

```
B --> C
```

```
B --> D
```

Mermaid 2

```
flowchart LR
```

```
A --o B
```

```
B --x C
```




```
flowchart LR
```

```
  A o--o B
```

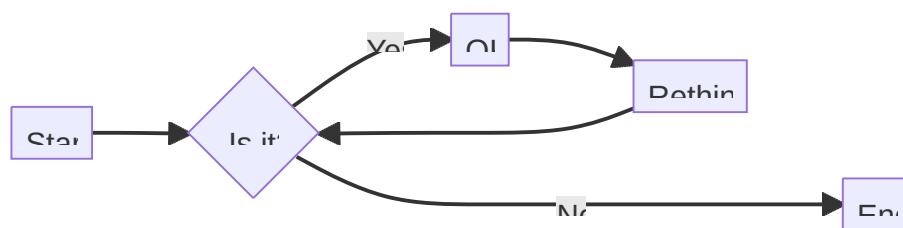
```
  B <--> C
```

```
  C x--x D
```





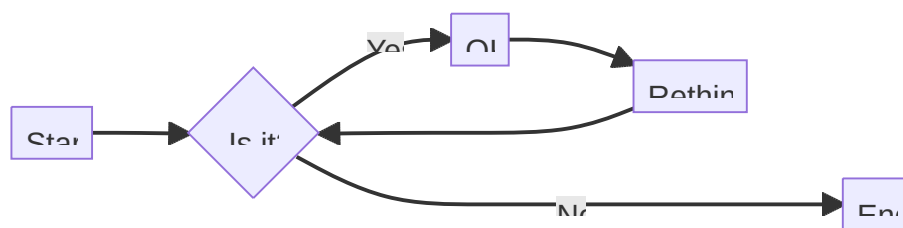
```
flowchart LR
    A[Start] --> B{Is it?}
    B -->|Yes| C[OK]
    C --> D[Rethink]
    D --> B
    B ---->|No| E[End]
```



-

```

flowchart LR
    A[Start] --> B{Is it?}
    B -- Yes --> C[OK]
    C --> D[Rethink]
    D --> B
    B -- No ----> E[End]
    
```

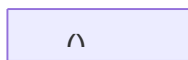


= .

	1	2	3
	—	—	—
	->	→	→
	===	===	===
	==>	===>	====>
	~.	~.	~.
	~.->	~.->	~.->

4499

```
flowchart LR
    id1["()"]
```

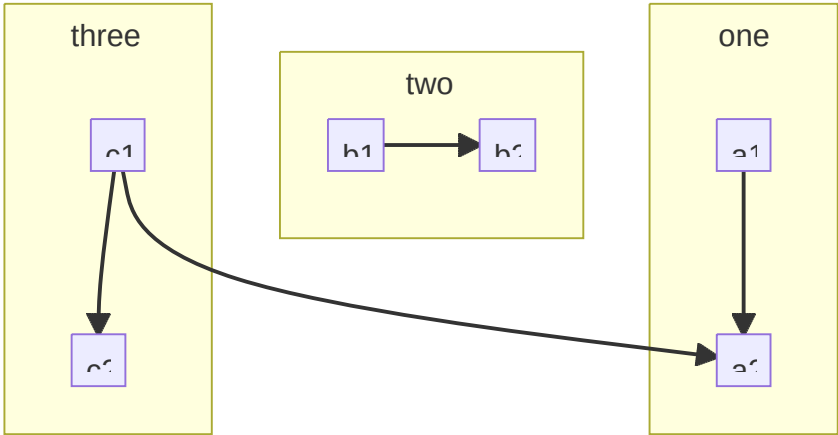
```
flowchart LR
```

```
A["A double quote:#quot;"] -->B["A dec char:#9829;"]
```



```
subgraph title
  graph definition
end
```

```
flowchart TB
  c1-->a2
  subgraph one
    a1-->a2
  end
  subgraph two
    b1-->b2
  end
  subgraph three
    c1-->c2
  end
end
```

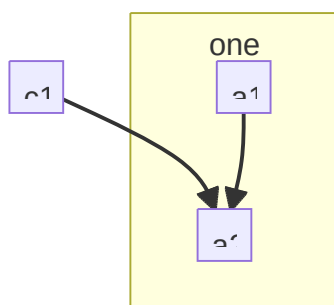


ID

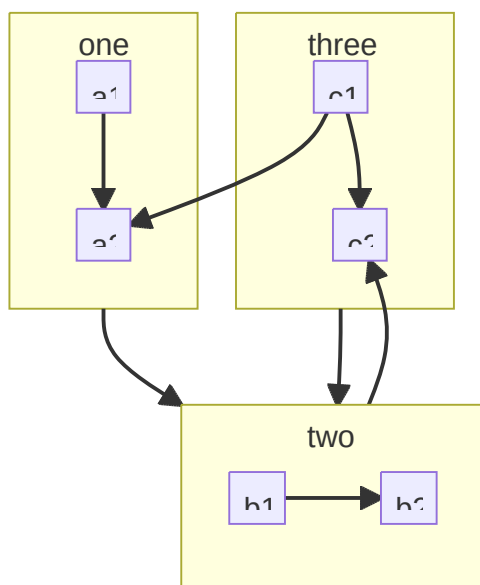
```

flowchart TB
  c1-->a2
  subgraph ide1 [one]
    a1-->a2
  end
end

```



```
flowchart TB
    c1-->a2
    subgraph one
        a1-->a2
    end
    subgraph two
        b1-->b2
    end
    subgraph three
        c1-->c2
    end
    one --> two
    three --> two
    two --> c2
```

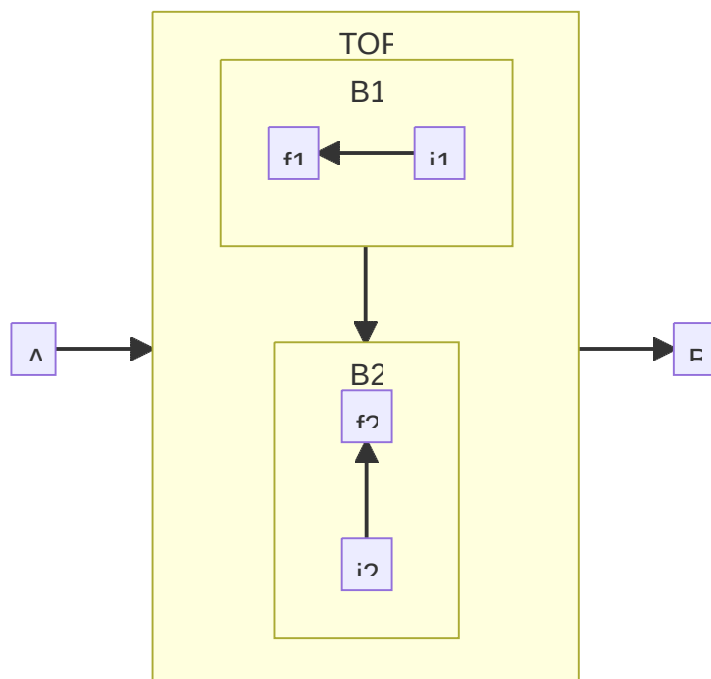



direction

```

flowchart LR
  subgraph TOP
    direction TB
    subgraph B1
      direction RL
      i1 -->f1
    end
    subgraph B2
      direction BT
      i2 -->f2
    end
  end
end
A --> TOP --> B
B1 --> B2

```



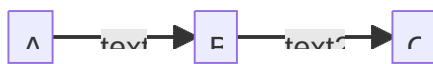
JavaScript URL

```
<script>  
  const callback = function () {  
    alert('A callback was triggered');  
  };  
</script>
```



%% Mermaid

```
flowchart LR
%% this is a comment A -- text --> B{node}
  A -- text --> B -- text2 --> C
```

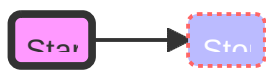


Mermaid

```

flowchart LR
    id1(Start)-->id2(Stop)
    style id1 fill:#f9f,stroke:#333,stroke-width:4px
    style id2 fill:#bbf,stroke:#f66,stroke-width:2px,color:#fff,stroke-d
asharray: 5 5

```

```

flowchart LR
    A:::someclass --> B --> C:::someclass
    classDef someclass fill:#f96

```



fontawesome

fontawesome Mermaid

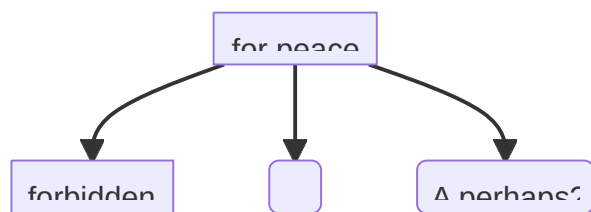
flowchart TD

```
B["fab:fa-twitter for peace"]
```

```
B-->C[fa:fa-ban forbidden]
```

```
B-->D(fa:fa-spinner)
```

```
B-->E(A fa:fa-camera-retro perhaps?)
```



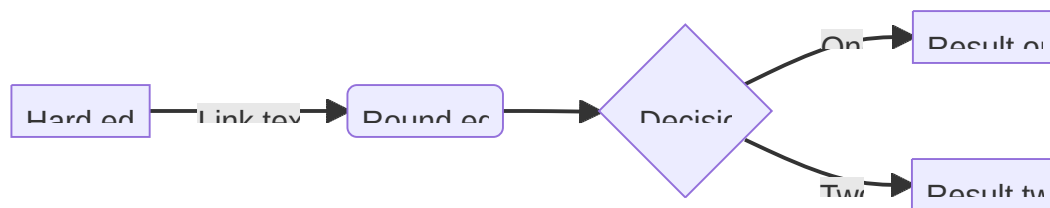
```
flowchart LR
```

```
  A[Hard edge] -->|Link text| B(Round edge)
```

```
  B --> C{Decision}
```

```
  C -->|One| D[Result one]
```

```
  C -->|Two| E[Result two]
```

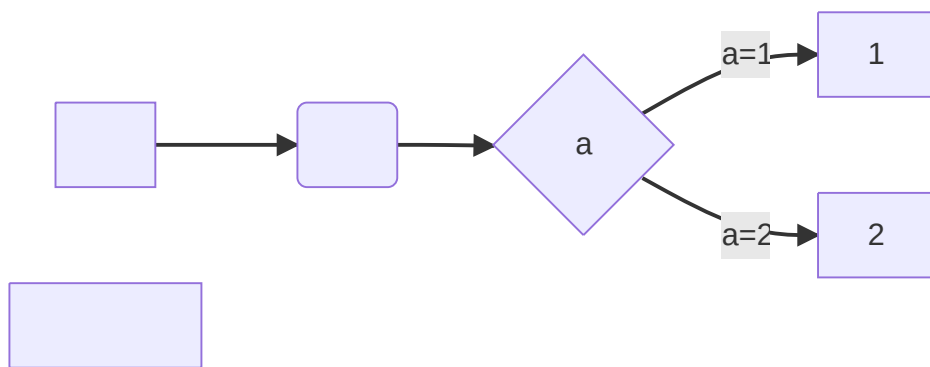


https://blog.csdn.net/qq_63585949/article/details/131544942

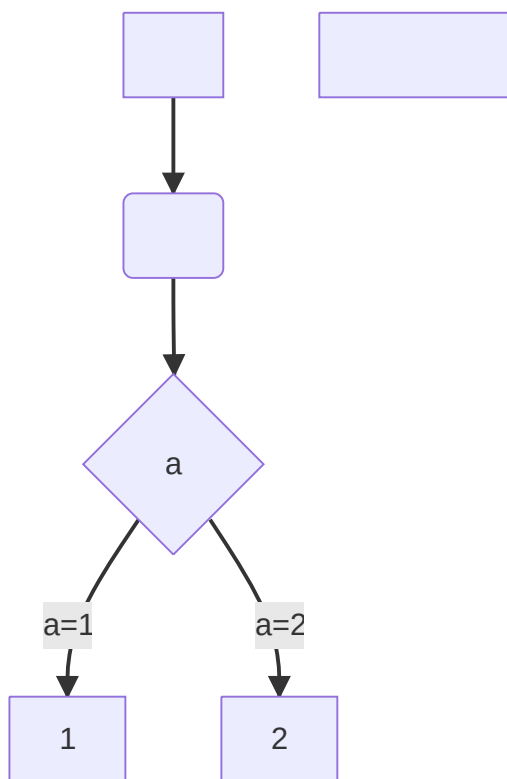
Markdown

Markdown

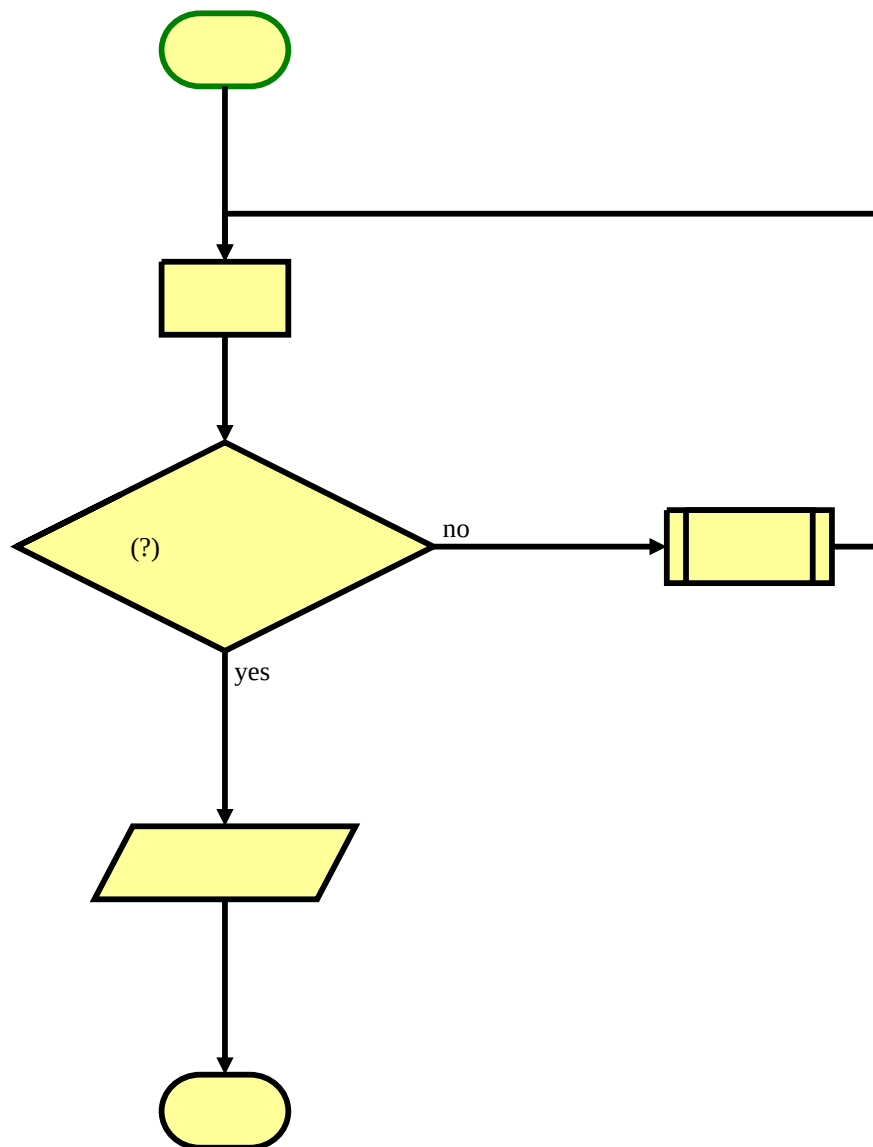
1



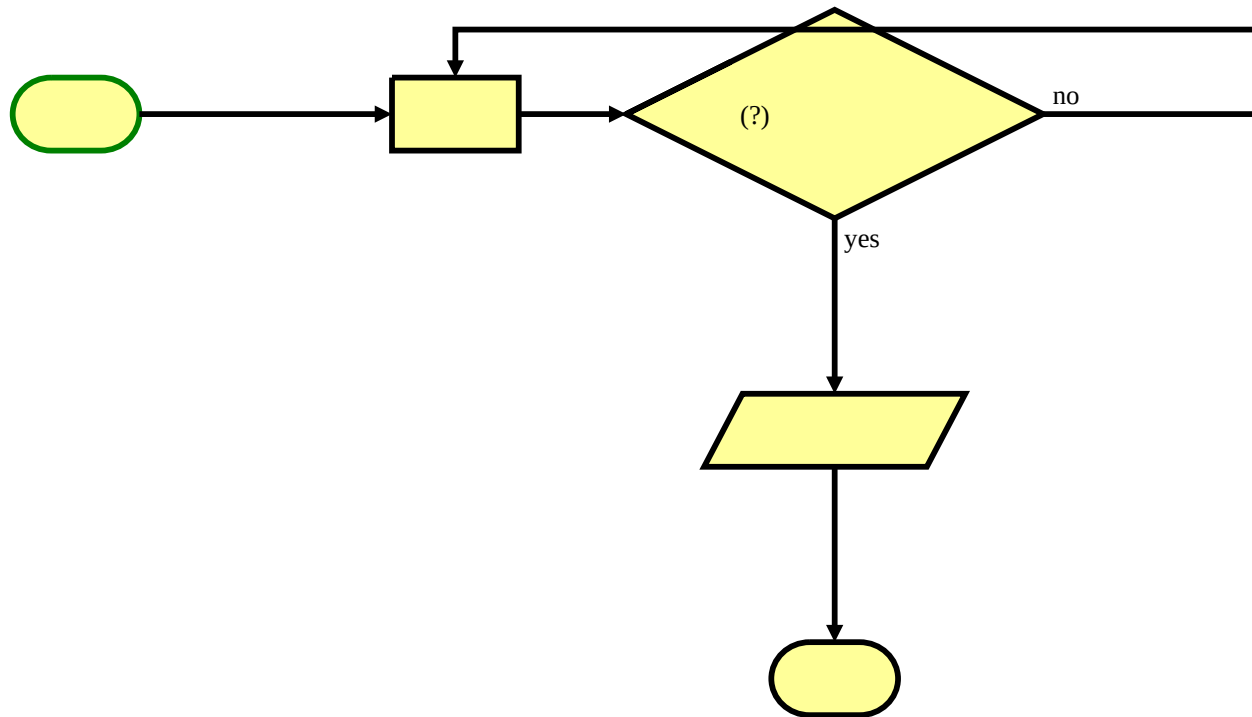
2



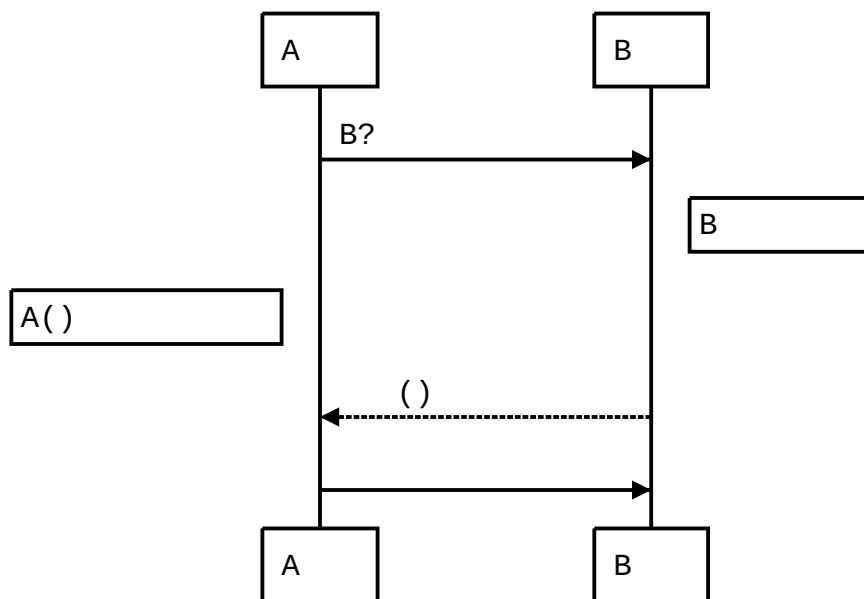
3



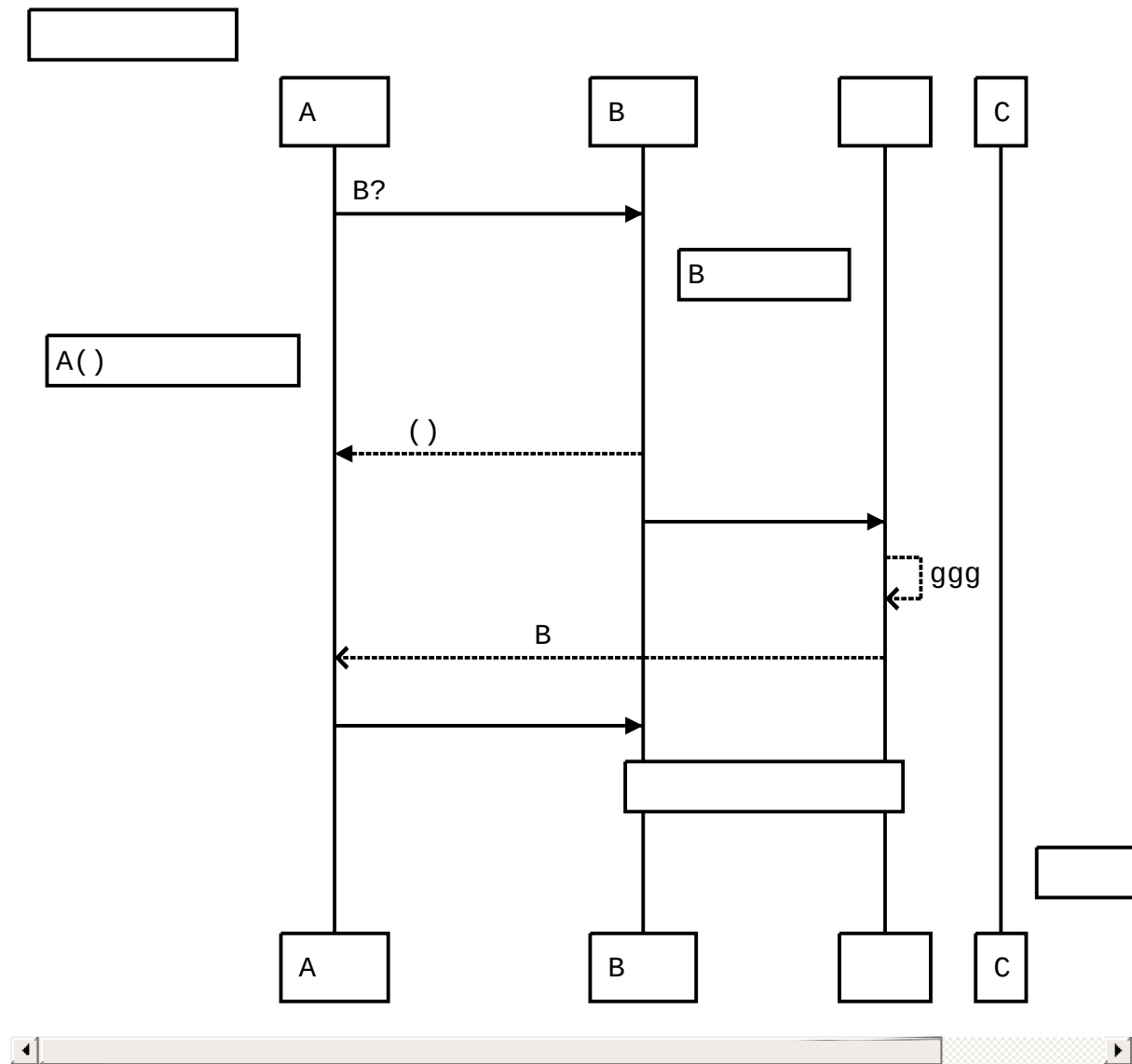
4



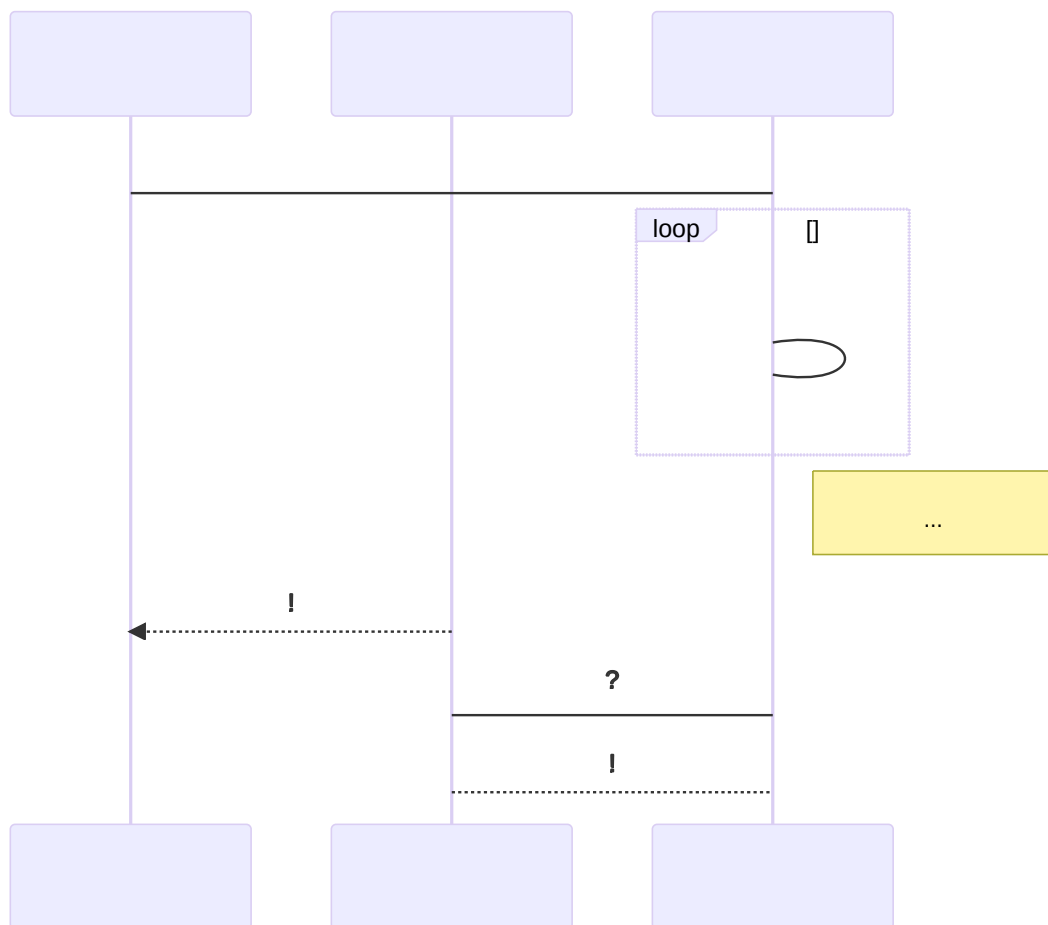
5UML



6UML



7UML



Markdown Mermaid ()

Markdown Mermaid () - (Sequence Diagram)

Mermaid

Mermaid

Mermaid JavaScript Markdown

Mermaid

- (Flowchart) :

graph

flowchart
- (Sequence Diagram) :

sequenceDiagram
- (Class Diagram) :

classDiagram
- (State Diagram) :

stateDiagram
- (Entity Relationship Diagram) :

erDiagram
- (User Journey Diagram) :

journey
- (Gantt Diagram) :

gantt
- (Pie Chart Diagram) :

pie
- (Quadrant Chart) :

quadrantChart
- (Requirement Diagram) :

requirementDiagram
- Gitgraph (Gitgraph Diagram) :

gitGraph
- (Mindmap) :

mindmap
- (Timeline Diagram) :

timeline

Mermaid

Mermaid

Mermaid

- Mermaid [Mermaid Live Editor](#)
- Mermaid Markdown [Typora](#)

Mermaid Mermaid

```
flowchart TD
```

```
  A[Christmas] -->|Get money| B(Go shopping)
```

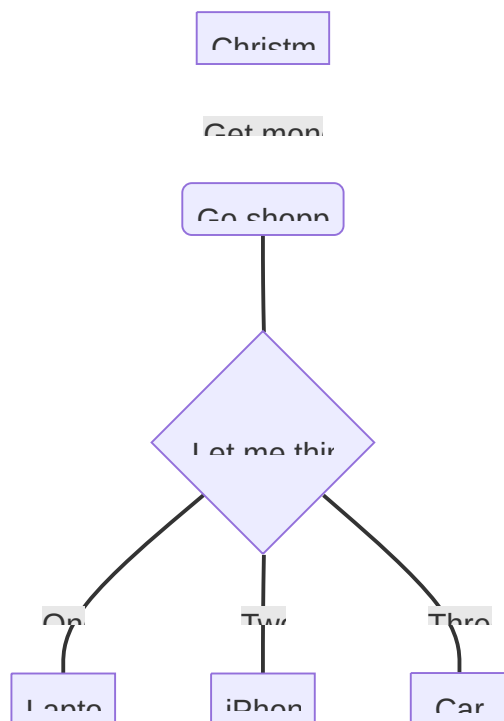
```
  B --> C{Let me think}
```

```
  C -->|One| D[Laptop]
```

```
  C -->|Two| E[iPhone]
```

```
  C -->|Three| F[fa:fa-car Car]
```

Mermaid Markdown Mermaid

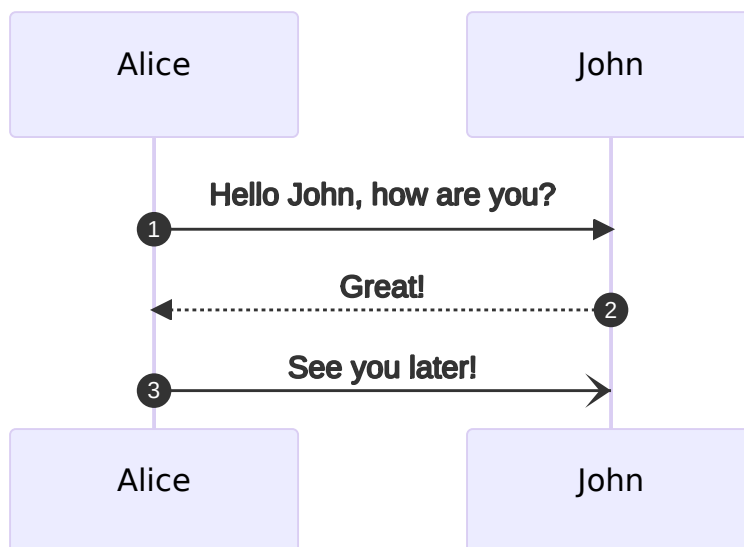


Mermaid

(Sequence Diagram)

Mermaid

```
sequenceDiagram
    Alice->>John: Hello John, how are you?
    John-->>Alice: Great!
    Alice-)John: See you later!
```

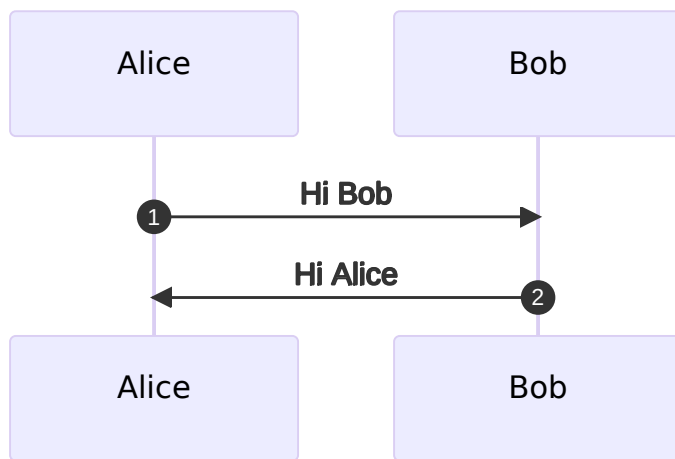


```
Mermaid      end      end  ([[]{}]
```

Participants

:

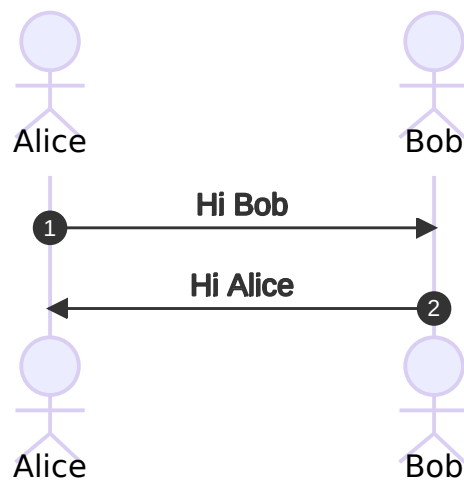
```
sequenceDiagram
    participant Alice
    participant Bob
    Alice->>Bob: Hi Bob
    Bob->>Alice: Hi Alice
```



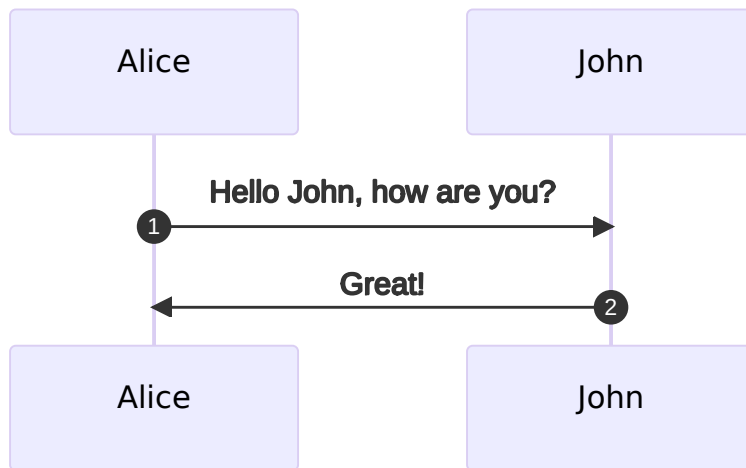
Actors

actor actor participant

```
sequenceDiagram
    actor Alice
    actor Bob
    Alice->>Bob: Hi Bob
    Bob->>Alice: Hi Alice
```



```
sequenceDiagram
    participant A as Alice
    participant J as John
    A->>J: Hello John, how are you?
    J->>A: Great!
```



```

box Aqua Group Description
... actors ...
end
box Group without description
... actors ...
end
box rgb(33,66,99)
... actors ...
end

```

transparent

```

box transparent Aqua
... actors ...
end

```

```

sequenceDiagram
    box Purple Alice & John
    participant A
    participant J
    end
    box Another Group
    participant B
    participant C
    end
    A->>J: Hello John, how are you?
    J->>A: Great!
    A->>B: Hello Bob, how is Charly ?
    B->>C: Hello Charly, how are you?

```

```

[Actor][Arrow][Actor]:Message text

```

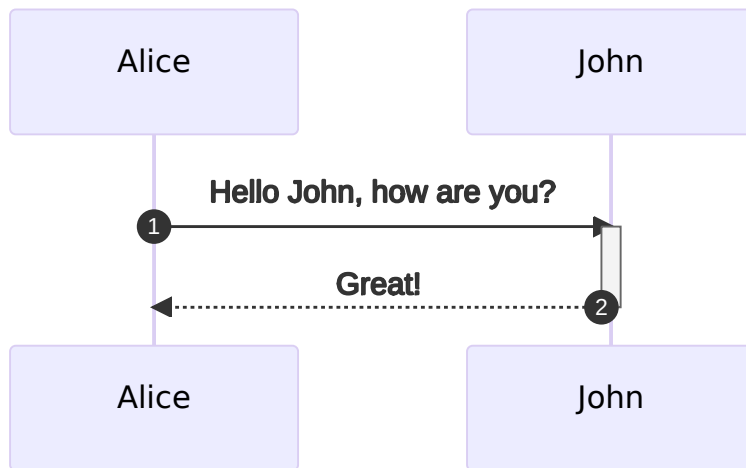
Mermaid

--	--

->	
-->	
>>	
-->>	
-x	x
--x	x
-)) (async)
--)) (async)

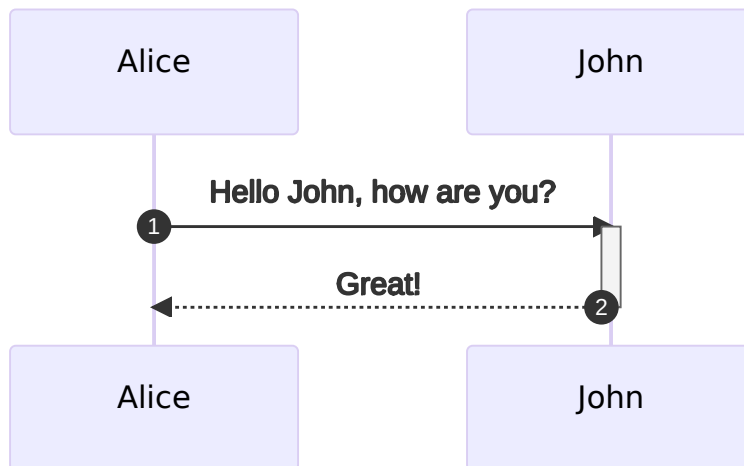
activate deactivate

```
sequenceDiagram
    Alice->>John: Hello John, how are you?
    activate John
    John-->>Alice: Great!
    deactivate John
```



+ -

```
sequenceDiagram
    Alice->>+John: Hello John, how are you?
    John-->>-Alice: Great!
```

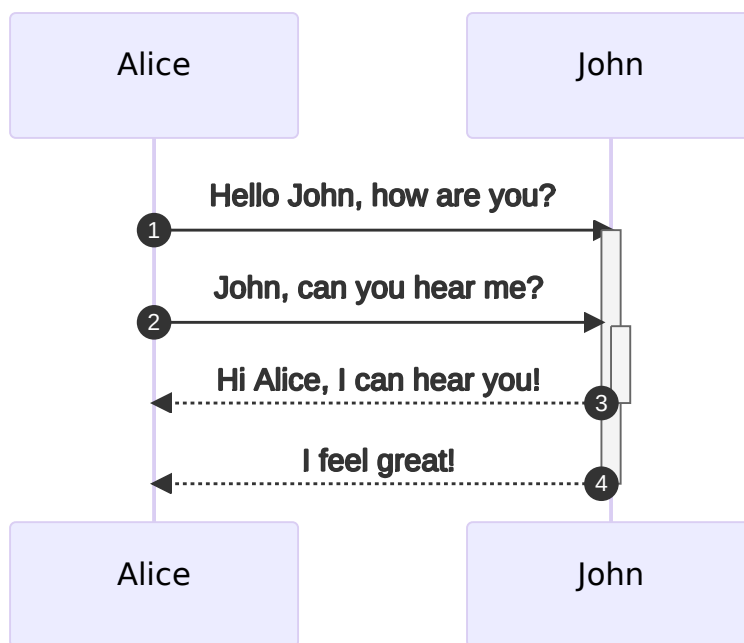
```
sequenceDiagram
```

```
  Alice->>+John: Hello John, how are you?
```

```
  Alice->>+John: John, can you hear me?
```

```
  John-->>-Alice: Hi Alice, I can hear you!
```

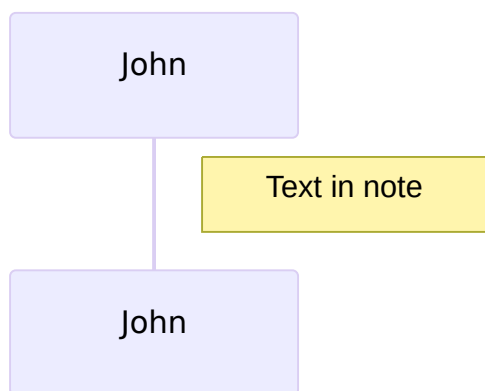
```
  John-->>-Alice: I feel great!
```



Note

Note [right of | left of | over] [Actor]: Text in note content

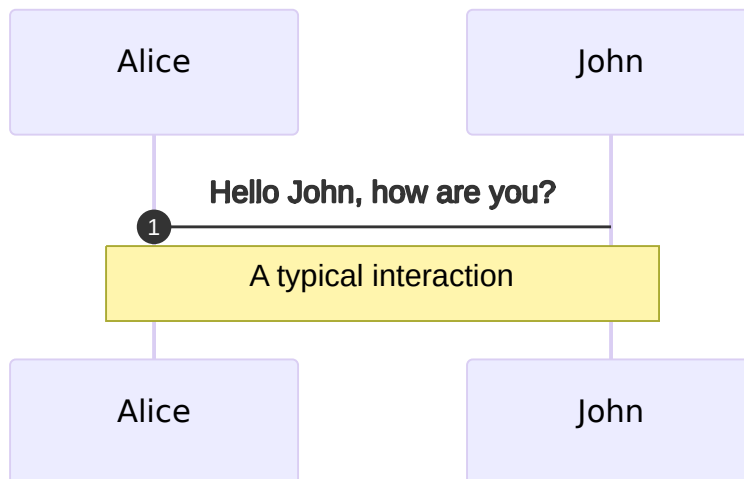
```
sequenceDiagram
    participant John
    Note right of John: Text in note
```



sequenceDiagram

Alice->>John: Hello John, how are you?

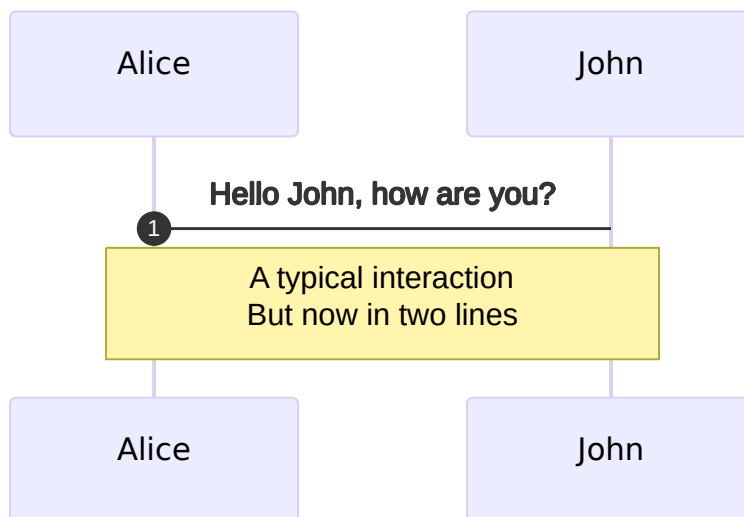
Note over Alice,John: A typical interaction



```
sequenceDiagram
```

```
    Alice->>John: Hello John, how are you?
```

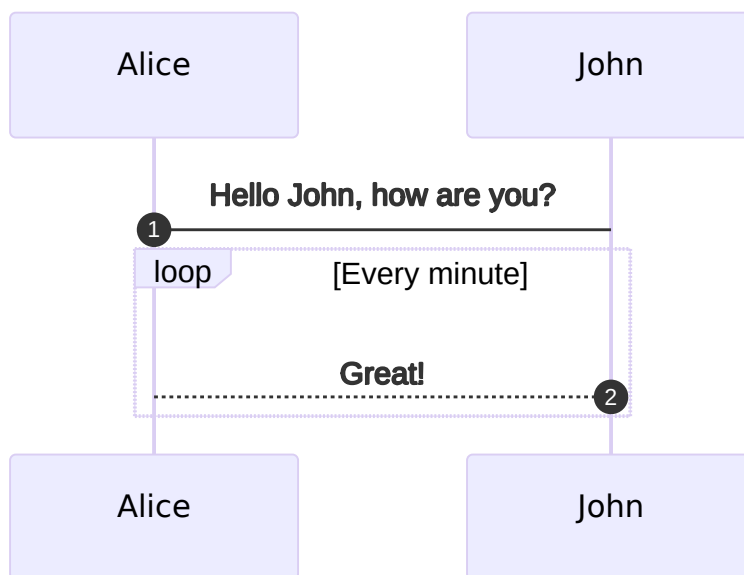
```
    Note over Alice,John: A typical interaction<br/>But now in two lines
```

loop

```
loop Loop text
... statements ...
end
```

```
sequenceDiagram
    Alice->>John: Hello John, how are you?
    loop Every minute
        John-->>Alice: Great!
    end
```

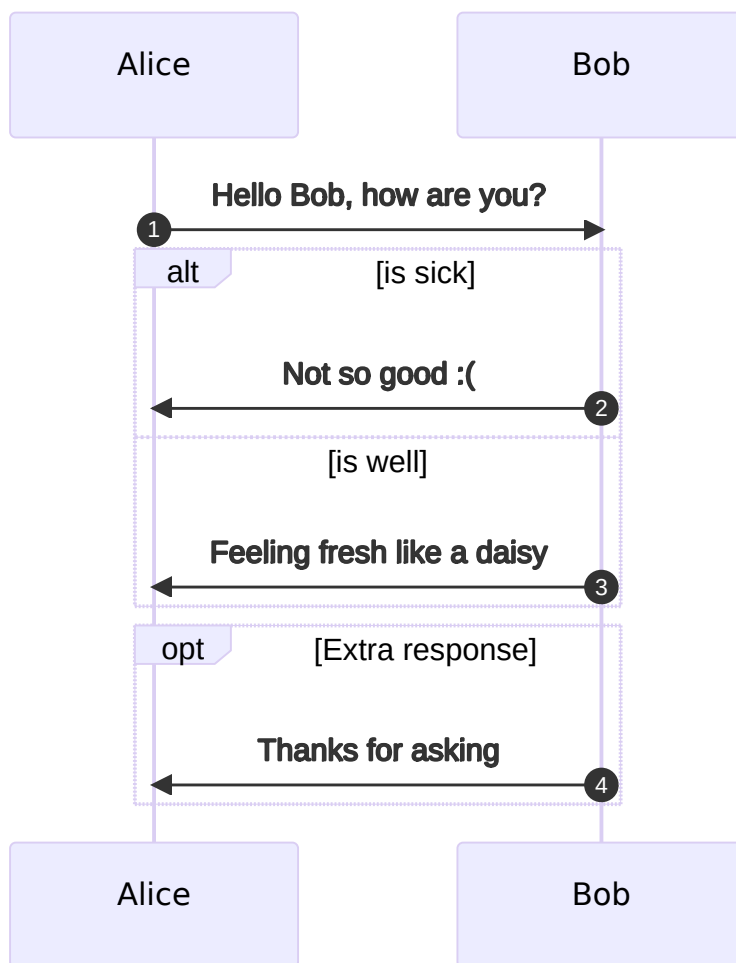


alt

```
alt Describing text
... statements ...
else
... statements ...
end
```

```
opt Describing text
... statements ...
end
```

```
sequenceDiagram
    Alice->>Bob: Hello Bob, how are you?
    alt is sick
        Bob->>Alice: Not so good :(
    else is well
        Bob->>Alice: Feeling fresh like a daisy
    end
    opt Extra response
        Bob->>Alice: Thanks for asking
    end
```

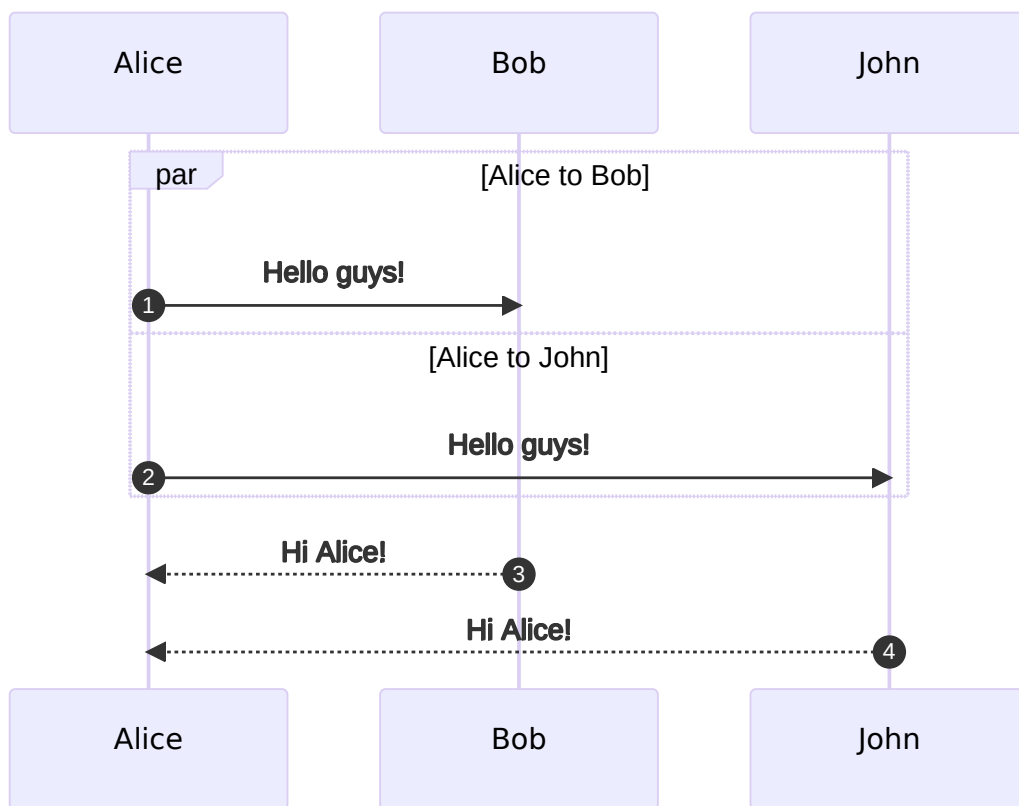


par

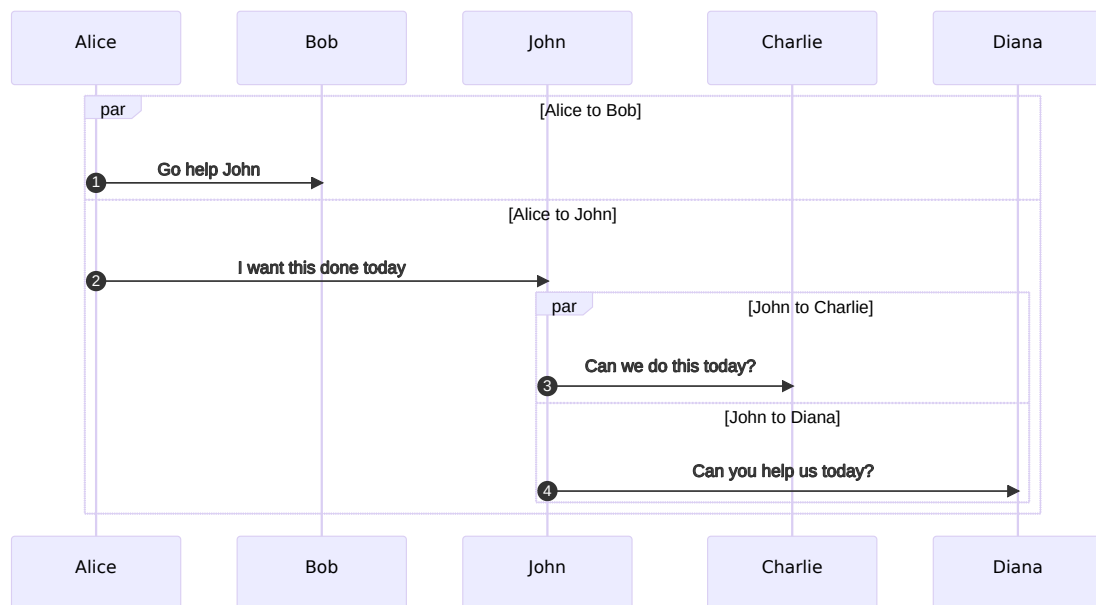
```
par [Action 1]
... statements ...
and [Action 2]
... statements ...
and [Action N]
... statements ...
end
```

sequenceDiagram

```
par Alice to Bob
    Alice->>Bob: Hello guys!
and Alice to John
    Alice->>John: Hello guys!
end
Bob-->>Alice: Hi Alice!
John-->>Alice: Hi Alice!
```

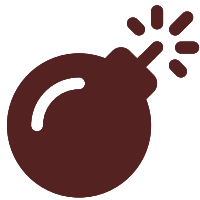


```
sequenceDiagram
    par Alice to Bob
        Alice->>Bob: Go help John
    and Alice to John
        Alice->>John: I want this done today
        par John to Charlie
            John->>Charlie: Can we do this today?
        and John to Diana
            John->>Diana: Can you help us today?
        end
    end
end
```

```
critical [Action that must be performed]
... statements ...
option [Circumstance A]
... statements ...
option [Circumstance B]
... statements ...
end
```

```
sequenceDiagram
    critical Establish a connection to the DB
        Service-->DB: connect
    option Network timeout
        Service-->Service: Log error
    option Credentials rejected
        Service-->Service: Log different error
    end
```



Syntax error in graph

mermaid version 8.13.0

```
sequenceDiagram
    critical Establish a connection to the DB
        Service-->DB: connect
    end
```


par

Break

()

```
break [something happened]
... statements ...
end
```

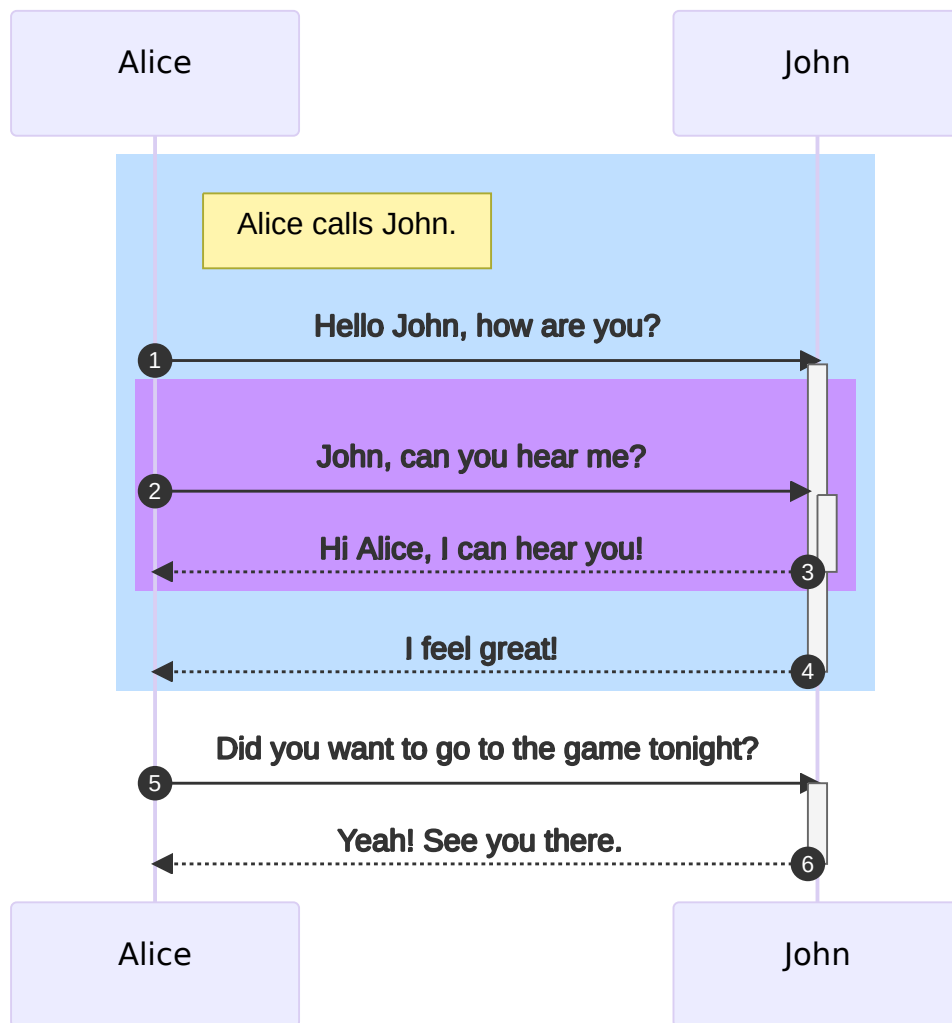
```
sequenceDiagram
    Consumer-->API: Book something
    API-->BookingService: Start booking process
    break when the booking process fails
    API-->Consumer: show failure
end
API-->BillingService: Start billing process
```



```
rect rgb(0, 255, 0)
... content ...
end
rect rgba(0, 0, 255, .1)
... content ...
end
```

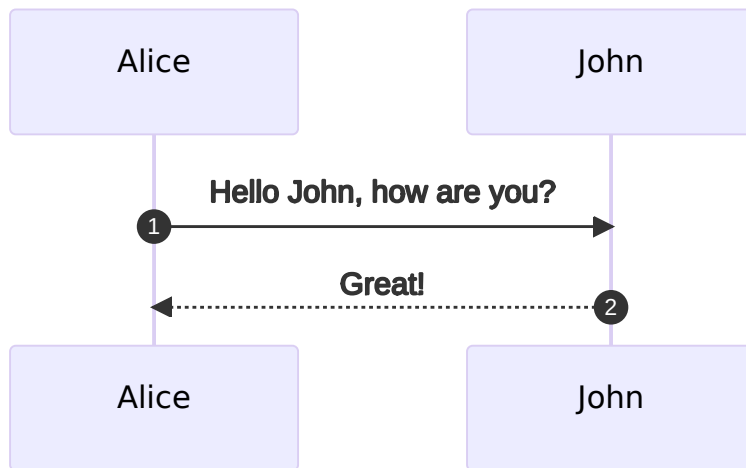
```
sequenceDiagram
    participant Alice
    participant John

    rect rgb(191, 223, 255)
    note right of Alice: Alice calls John.
    Alice->>+John: Hello John, how are you?
    rect rgb(200, 150, 255)
    Alice->>+John: John, can you hear me?
    John-->>-Alice: Hi Alice, I can hear you!
    end
    John-->>-Alice: I feel great!
    end
    Alice ->>+ John: Did you want to go to the game tonight?
    John -->>- Alice: Yeah! See you there.
```



%%

```
sequenceDiagram
    Alice->>John: Hello John, how are you?
    %% this is a comment
    John-->>Alice: Great!
```

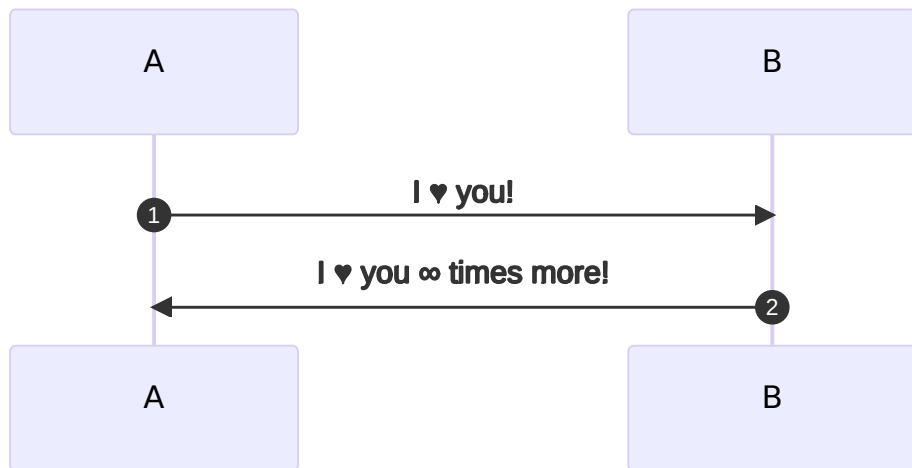



HTML

```
sequenceDiagram
```

```
A->>B: I #9829; you!
```

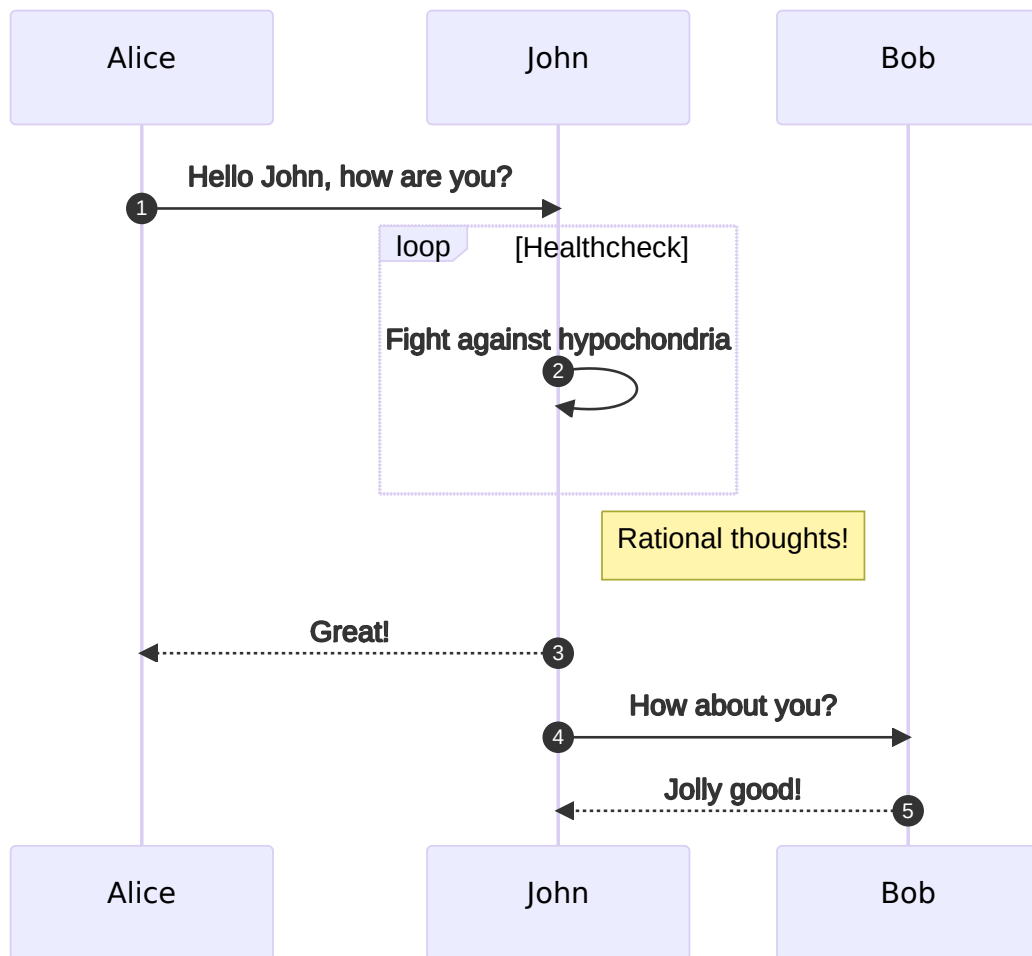
```
B->>A: I #9829; you #infin; times more!
```



```
<script>
  mermaid.initialize({ sequence: { showSequenceNumbers: true } });
</script>
```

autonumber

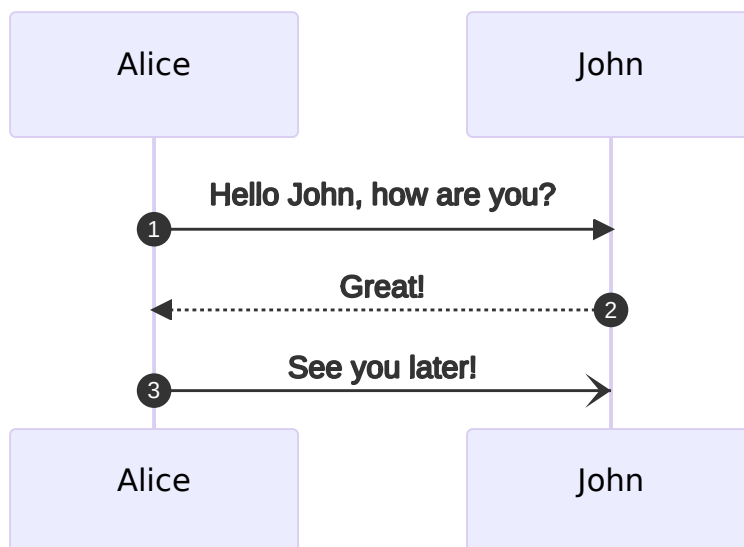
```
sequenceDiagram
    autonumber
    Alice->>John: Hello John, how are you?
    loop Healthcheck
        John->>John: Fight against hypochondria
    end
    Note right of John: Rational thoughts!
    John-->>Alice: Great!
    John->>Bob: How about you?
    Bob-->>John: Jolly good!
```



web repo wiki

```
link <actor>: <link-label> @ <link-url>
```

```
sequenceDiagram
    participant Alice
    participant John
    link Alice: Dashboard @ https://dashboard.contoso.com/alice
    link Alice: Wiki @ https://wiki.contoso.com/alice
    link John: Dashboard @ https://dashboard.contoso.com/john
    link John: Wiki @ https://wiki.contoso.com/john
    Alice->>John: Hello John, how are you?
    John-->>Alice: Great!
    Alice-)John: See you later!
```



1

json

```
links <actor>: <json-formatted link-name link-url pairs>
```

```
sequenceDiagram
    participant Alice
    participant John
    links Alice: {"Dashboard": "https://dashboard.contoso.com/alice", "Wiki": "https://wiki.contoso.com/alice"}
    links John: {"Dashboard": "https://dashboard.contoso.com/john", "Wiki": "https://wiki.contoso.com/john"}
    Alice->>John: Hello John, how are you?
    John-->>Alice: Great!
    Alice-)John: See you later!
```

```
sequenceDiagram
    participant Alice
    participant John
    links Alice: {"Dashboard": "https://dashboard.contoso.com/alice", "Wiki": "https://wiki.contoso.com/alice"}
    links John: {"Dashboard": "https://dashboard.contoso.com/john", "Wiki": "https://wiki.contoso.com/john"}
    Alice->>John: Hello John, how are you?
    John-->>Alice: Great!
    Alice-)John: See you later!
```

css

```
src/themes/sequence.scss
```

actor	
text.actor	
actor-line	
messageLine0	

messageLine1	
messageText	
labelBox	
labelText	
loopText	
loopLine	
note	
noteText	

```
body {
  background: white;
}

.actor {
  stroke: #ccccff;
  fill: #ececff;
}
text.actor {
  fill: black;
  stroke: none;
  font-family: Helvetica;
}

.actor-line {
  stroke: grey;
}

.messageLine0 {
  stroke-width: 1.5;
  stroke-dasharray: '2 2';
  marker-end: 'url(#arrowhead)';
  stroke: black;
}

.messageLine1 {
  stroke-width: 1.5;
  stroke-dasharray: '2 2';
  stroke: black;
```

```

}

#arrowhead {
  fill: black;
}

.messageText {
  fill: black;
  stroke: none;
  font-family: 'trebuchet ms', verdana, arial;
  font-size: 14px;
}

.labelBox {
  stroke: #ccccff;
  fill: #ececff;
}

.labelText {
  fill: black;
  stroke: none;
  font-family: 'trebuchet ms', verdana, arial;
}

.loopText {
  fill: black;
  stroke: none;
  font-family: 'trebuchet ms', verdana, arial;
}

.loopLine {
  stroke-width: 2;
  stroke-dasharray: '2 2';
  marker-end: 'url(#arrowhead)';
  stroke: #ccccff;
}

.note {
  stroke: #decc93;
  fill: #fff5ad;
}

.noteText {
  fill: black;
  stroke: none;
  font-family: 'trebuchet ms', verdana, arial;
  font-size: 14px;
}

```

json mermaid.sequenceConfigmermaid.sequenceConfig

```
mermaid.sequenceConfig = {
  diagramMarginX: 50,
  diagramMarginY: 10,
  boxTextMargin: 5,
  noteMargin: 10,
  messageMargin: 35,
  mirrorActors: true,
};
```

mirrorActors	/	false
bottomMarginAdj	css	1
actorFontSize		14
actorFontFamily		"Open Sans", sans-serif
actorFontWeight		"Open Sans", sans-serif
noteFontSize		14
noteFontFamily		"trebuchet ms", verdana, arial
noteFontWeight		"trebuchet ms", verdana, arial
noteAlign		center
messageFontSize		16
messageFontFamily		"trebuchet ms", verdana, arial
messageFontWeight		"trebuchet ms", verdana, arial

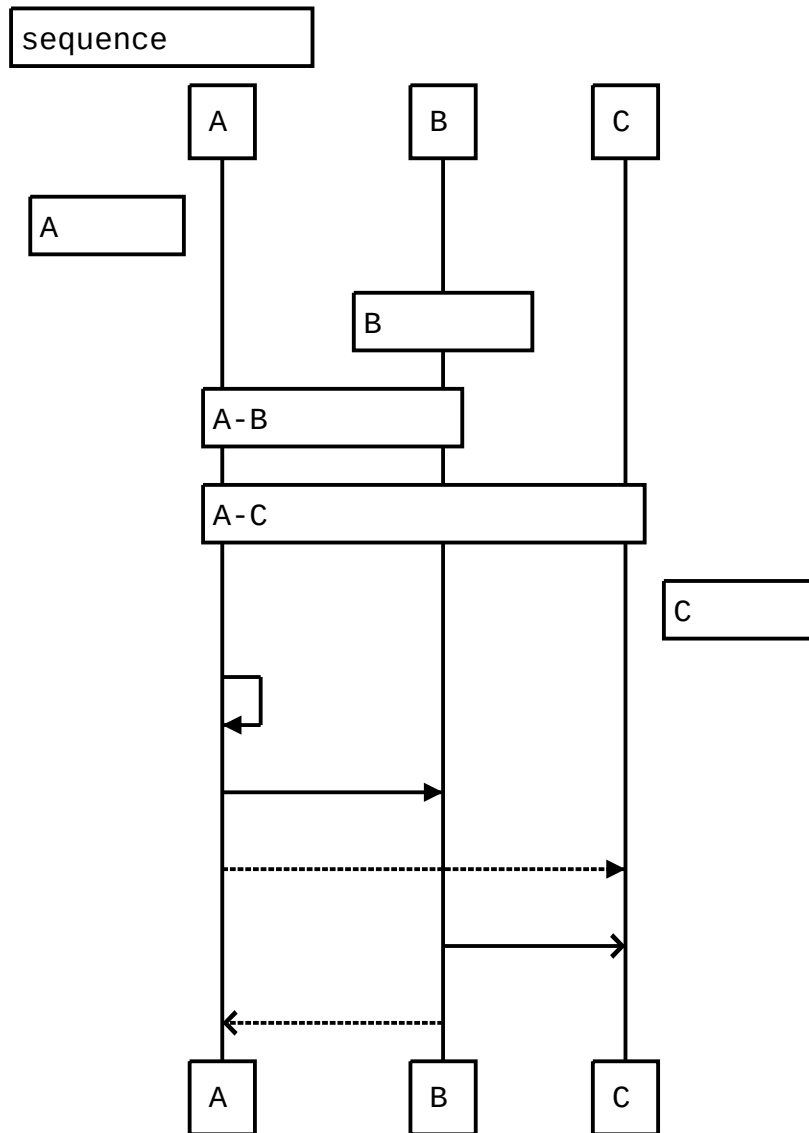
https://blog.csdn.net/qq_63585949/article/details/131580210

Markdown

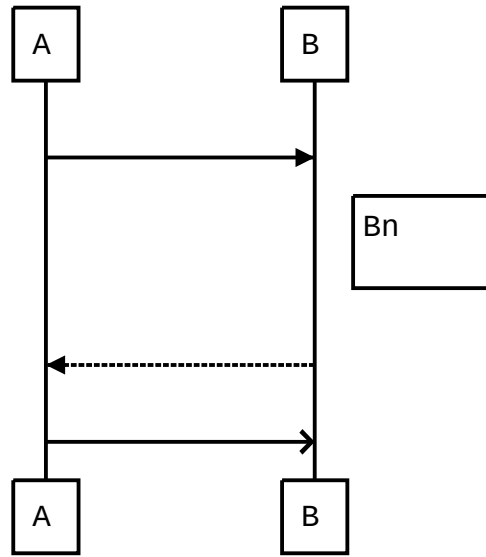
Sequence

- title
- participant
- note
- - left of
 - right of
 - over
- {actor}
- - ->
 - →
 - ->>
 - →>>

example



sequence AABBBCCABC



AABBBn

Markdown

Markdown

8



Syntax error in graph
mermaid version 8.13.0

- Markdown Mermaid () -

(Flowchart) https://blog.csdn.net/qq_63585949/article/details/131544942