

```
1
2 public class GrantTypeAuthenticationConverter implements AuthenticationConverter {
3
4     public static final String GRANT_TYPE = "grant_type";
5
6     @Nullable
7     @Override
8     public Authentication convert( HttpServletRequest request ) {
9
10         // grant_type (REQUIRED)
11         String grantType = request.getParameter( GRANT_TYPE );
12         if ( !"urn:ietf:params:oauth:grant-type:custom_code".equals( grantType ) ) {
13             return null;
14         }
15
16         Authentication clientPrincipal = SecurityContextHolder.getContext()
17                                     .getAuthentication();
18
19         MultiValueMap< String , String > parameters = getParameters( request );
20
21         // code (REQUIRED)
22         String code = parameters.getFirst( OAuth2ParameterNames.CODE );
23         if ( !StringUtils.hasText( code ) || parameters.get( OAuth2ParameterNames.CODE ).size() != 1 ) {
24             throw new OAuth2AuthenticationException( OAuth2ErrorCodes.INVALID_REQUEST );
25         }
26
27         Map< String, Object > additionalParameters = new HashMap<>();
28
29         parameters.forEach( ( key , value ) -> {
30             if ( !key.equals( OAuth2ParameterNames.GRANT_TYPE ) && !key.equals(
31                 OAuth2ParameterNames.CLIENT_ID ) && !key.equals( OAuth2ParameterNames.CODE ) ) {
32                 additionalParameters.put( key , value.get( 0 ) );
33             }
34         } );
35         return new OAuth2AuthorizationCodeAuthenticationToken( code , clientPrincipal , additionalParameters );
36     }
37
38     private MultiValueMap< String, String > getParameters( HttpServletRequest request ) {
39
40         Map< String, String[] > parameterMap = request.getParameterMap();
41
42         MultiValueMap< String, String > parameters = new LinkedMultiValueMap<>( parameterMap.size() );
43
44         parameterMap.forEach( ( key , values ) -> {
45             if ( values.length > 0 ) {
46                 for ( String value : values ) {
47                     parameters.add( key , value );
48                 }
49             }
50         } );
51         return parameters;
52     }
53 }
54
```