KAZAKH KBTU BRITISH
TECHNICAL
UNIVERSITY

WEB DEVELOPMENT

Lesson 13

CRUD using DRF

Wrapping API views

Wrapping API views

1. @api_view — FBV

```
from rest_framework.decorators import api_view
```

2. APIView — CBV

```
from rest_framework.views import APIView
```

- receive **Request** instance in view
- add context to **Response**
- method not allowed
- parsing error

Function Based Views

Rewriting our API using @api_view

```
from rest_framework.decorators import api_view
```

Class Based Views

Rewriting our API using class-based views

```
from rest_framework.views import APIView
```

Requests and Responses

Status code

```
from rest_framework import status
```

return Response(data, **status=status.HTTP_200_OK**)

Generic class-based views and Mixins

```
from rest_framework import mixins
from rest_framework import generics
```

```
class StudentList(generics.ListCreateAPIView):
    queryset = Student.objects.all()
    serializer_class = StudentSerializer
```

related_name attribute in models

```
class Product(models.Model):
  category = models.ForeignKey(Category,
                               on_delete=models.CASCADE,
                               related_name='products')
...
```

```
category.products.all()
```

instead of **product_set**

Create product with category

```
category_id = serializers.IntegerField(write_only=True)
```

Get category with its products

1. StringRelatedField
2. PrimaryKeyRelatedField
3. Nested objects

Authentication & Permissions

DRF Authentication

1. BasicAuthentication
2. TokenAuthentication
3. SessionAuthentication
4. *Third party auth*

DRF Authentication

```
REST_FRAMEWORK = {
  'DEFAULT_AUTHENTICATION_CLASSES': (
    'rest_framework.authentication.BasicAuthentication',
    ‘rest_framework.authentication.SessionAuthentication',
    ‘rest_framework.authentication.TokenAuthentication’,
  )
}
```

Django Rest Framework JWT

http://jpadilla.github.io/django-rest-framework-jwt/

What is JWT?

What is a JSON Web Token?

A JSON web token, or JWT (“jot”) for short, is a standardized, optionally validated and/or encrypted container format that is used to securely transfer information between two parties.

What is a JSON Web Token?

JWTs consist of three parts separated by dots (.), which are:

- header
- payload
- signature

JWT typically looks like the following:

```
xxxxx.yyyyy.zzzzz
```

Header

The header typically consists of two parts: the type of the token, which is JWT, and the hashing algorithm such as HMAC SHA256 or RSA.

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

this JSON is Base64Url encoded to form the first part of the JWT.

Payload

The second part of the token is the payload, which contains the claims. Claims are statements about an entity (typically, the user) and additional metadata.

```
{
  "sub": "1234567890",
  "name": “KBTU FIT",
  "admin": true
}
```

The payload is then Base64Url encoded to form the second part of the JWT.

Signature

To create the signature part you have to take the encoded header, the encoded payload, a secret, the algorithm specified in the header, and sign that.

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret)
```

The signature is used to verify that the sender of the JWT is who it says it is and to ensure that the message was’t changed in the way.

Putting all together

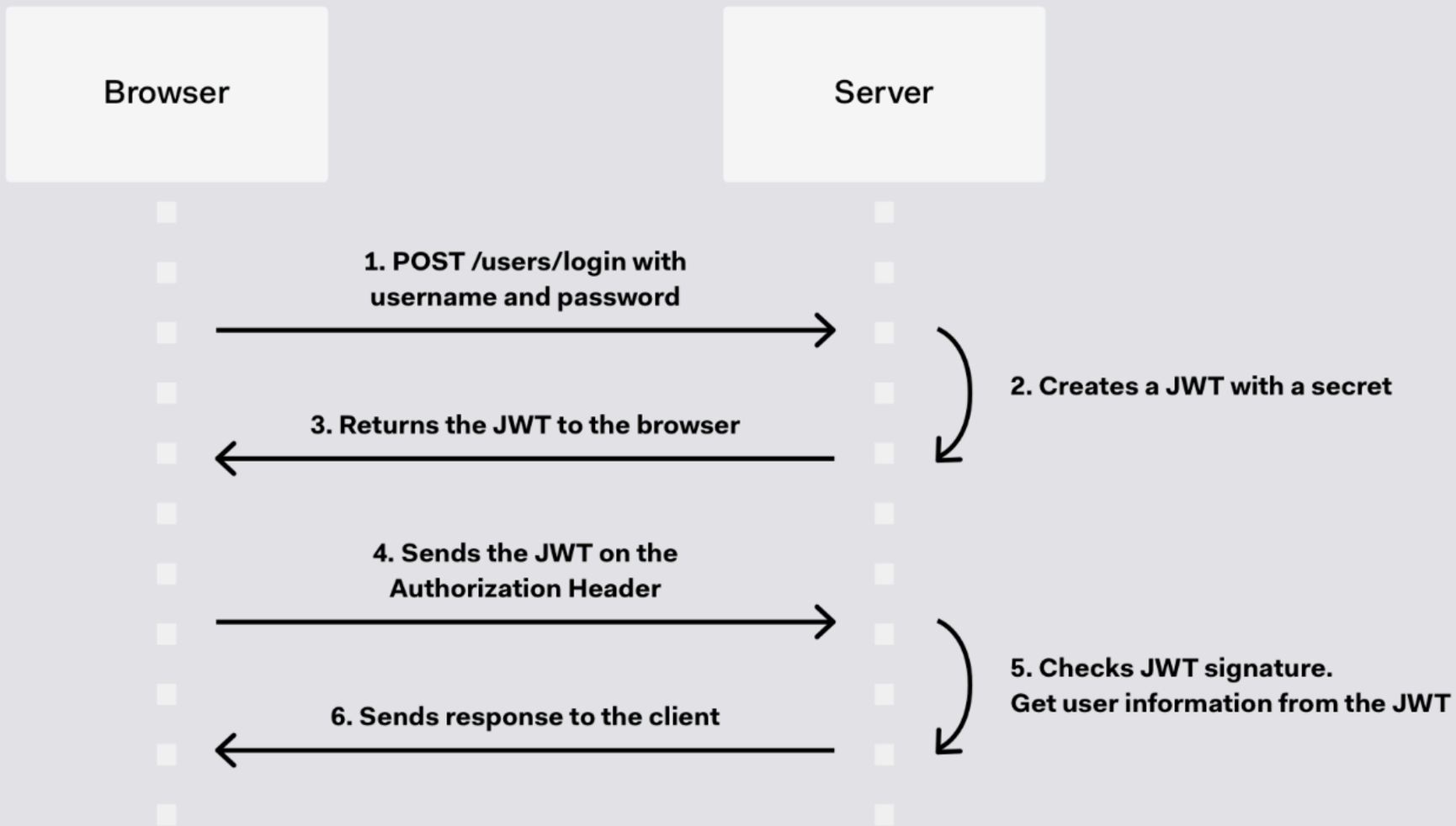
The following shows a JWT that has the previous header and payload encoded and it is signed with a secret.

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4
gRG9lIiwiaXNTb2NpYWwiOnRydWV9.
4pcPyMD09olPSyXnrXCjTwXyr4BsezdI1AVTmud2fU4
```

Access to resource

header of each http request must contain

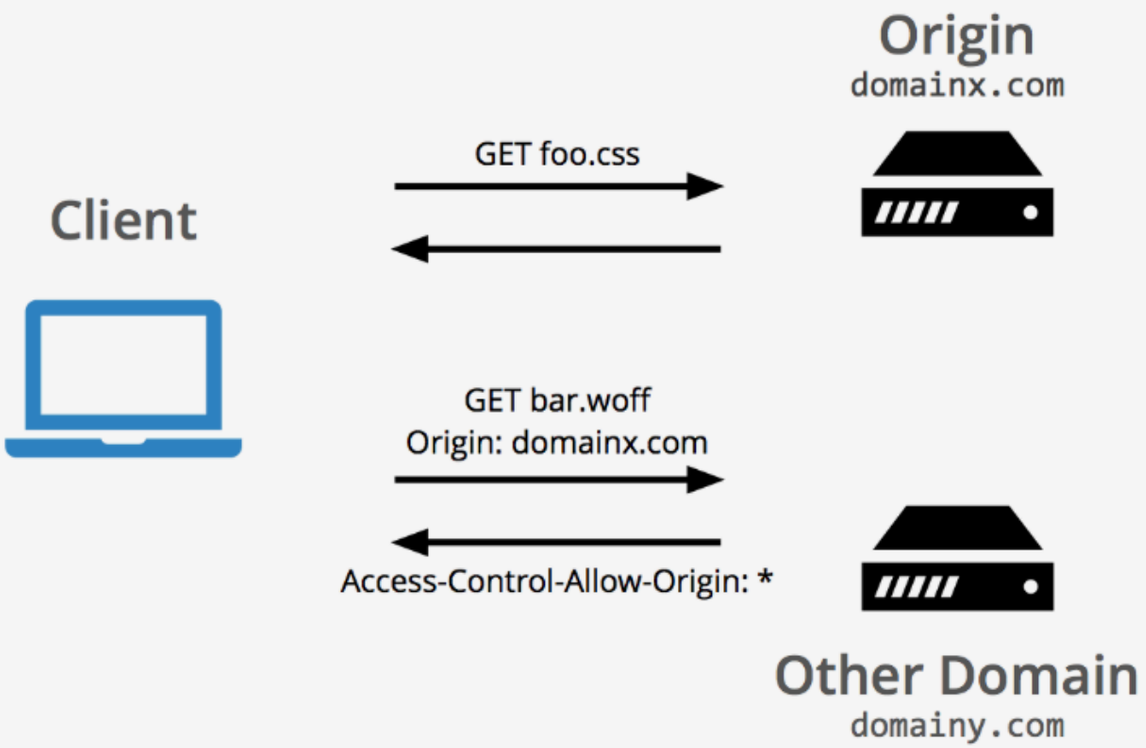
```
Authorization: JWT <token>
```


Browser
Server
1. POST /users/login with username and password
2. Creates a JWT with a secret
3. Returns the JWT to the browser
4. Sends the JWT on the Authorization Header
5. Checks JWT signature. Get user information from the JWT
6. Sends response to the client

Cross-Origin Resource Sharing (CORS)

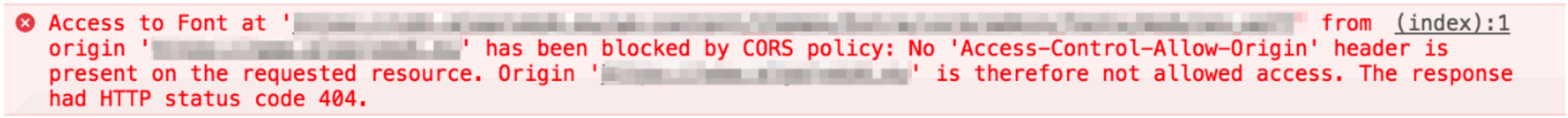
https://github.com/ottoyiu/django-cors-headers

Cross-Origin Resource Sharing (CORS)



CORS

Cross-Origin Resource Sharing (CORS)



Questions?