

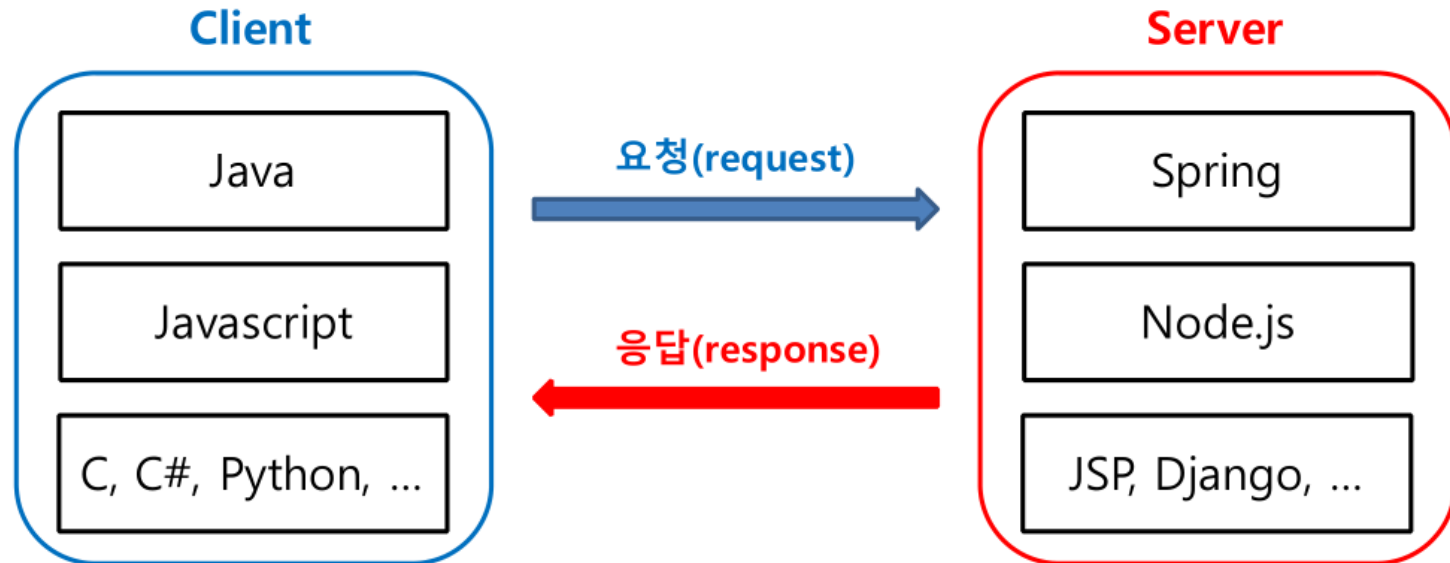
데이터 요청

1. 요청 메소드 및 데이터 형태
2. AJAX

■ 요청 메소드

● GET / POST

- Query Parameter / Form Data / JSON Data



■ 요청 메소드

● GET / POST

- Query Parameter / Form Data / JSON Data



■ 요청 데이터 형태

● GET / POST

- Query Parameter (GET)

▼ General

Request URL: `http://127.0.0.1:8000/first/req/get/?a=100&b=200&c=300`

Request Method: GET

Status Code: 200 OK

Remote Address: 127.0.0.1:8000

Referrer Policy: same-origin

▶ Response Headers (7)

▶ Request Headers (13)

▼ Query String Parameters view source view URL-encoded

a: 100

b: 200

c: 300

- Form Data (POST)

▼ General

Request URL: `http://127.0.0.1:8000/first/req/post/`

Request Method: POST

Status Code: 200 OK

Remote Address: 127.0.0.1:8000

Referrer Policy: same-origin

▶ Response Headers (7)

▶ Request Headers (17)

▼ Form Data view source view decoded

a: 10

b: 20

c: 30

- JSON Data (POST)

▼ General

Request URL: `http://127.0.0.1:8000/first/req/json/`

Request Method: POST

Status Code: 200 OK

Remote Address: 127.0.0.1:8000

Referrer Policy: same-origin

▶ Response Headers (7)

▶ Request Headers (17)

▼ Request Payload view source

{a: 10, b: 20, c: 30}

a: 10

b: 20

c: 30

■ 요청 데이터 형태

- Query Parameter → GET + ?

ex) /req/get/?desc=xyz&page=3&title=qwe

- Form Data → POST + FormData()

ex) /req/post/

```
const formData = new FormData();  
formData.append('a', 10);  
formData.append('b', 20);  
formData.append('c', 30);
```

- JSON Data → POST + JSON String + application/json

ex) /req/json/

```
const data = { 'a': 10, 'b': 20, 'c': 30 };  
  
...  
  
{  
  method: 'post',  
  headers: { 'Content-Type': 'application/json' },  
  body: JSON.stringify(data)  
};
```

■ AJAX

● JavaScript Old API

```
const xhr = new XMLHttpRequest();  
xhr.onreadystatechange = 콜백 함수;  
xhr.open('메소드', '주소');  
xhr.send(formData);
```

● JavaScript New API

```
const ajax = fetch('주소', { 옵션 });  
ajax.then(콜백 함수);
```

● jQuery

```
<script src='CDN 주소'></script>  
  
$.ajax({  
  url: '주소',  
  type: '메소드',  
  data: Query Parameter  
  success: 콜백 함수  
});
```

● Axios

```
<script src='CDN 주소'></script>  
  
const data = axios({  
  url: '/first/req/get/',  
  method: 'get',  
  params: Query Parameter  
});  
data.then(콜백 함수);
```

■ AJAX

● JavaScript Old API (GET)

```
<script>
  const xhr = new XMLHttpRequest();
  xhr.onreadystatechange = callback;
  xhr.open('get', '/first/req/get/?a=100&b=200&c=300');
  xhr.send(null);

  function callback() {
    if(xhr.readyState == 4) {
      if(xhr.status == 200) {
        let result = xhr.response;
        console.log(result);
      }
    }
  }
</script>
```

■ AJAX

● JavaScript Old API (POST)

```
<script>
  const xhr = new XMLHttpRequest();
  xhr.onreadystatechange = callback;
  xhr.open('post', '/first/req/post/');

  const formData = new FormData();
  formData.append('a', 100);
  formData.append('b', 200);
  formData.append('c', 300);

  xhr.send(formData);

  function callback() {
    if (xhr.readyState == 4) {
      if (xhr.status == 200) {
        let result = xhr.response;
        console.log(result);
      }
    }
  }
</script>
```


■ AJAX

● JavaScript New API (GET)

```
<script>
  const ajax = fetch('/first/req/get/?a=10&b=20&c=30');
  ajax.then(function(response) {
    return response.text();
  }).then(function(result) {
    console.log(result);
  });
</script>
```

■ AJAX

● JavaScript New API (POST - Form Data)

```
<script>
  const formData = new FormData();
  formData.append('a', 10);
  formData.append('b', 20);
  formData.append('c', 30);

  const ajax = fetch('/first/req/post/', {
    method: 'post',
    body: formData
  });
  ajax.then(function(response) {
    return response.text();
  }).then(function(result) {
    console.log(result);
  });
</script>
```

■ AJAX

● JavaScript New API (POST - JSON Data)

```
<script>
  const formData = new FormData();
  formData.append('a', 10);
  formData.append('b', 20);
  formData.append('c', 30);

  const ajax = fetch('/first/req/json/', {
    method: 'post',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(Object.fromEntries(formData))
  });
  ajax.then(function(response) {
    return response.json();
  }).then(function(result) {
    console.log(result);
  });
</script>
```

■ AJAX

● jQuery (GET)

```
<script src='http://code.jquery.com/jquery-3.1.1.min.js'></script>
<script>
  $.ajax({
    url: `/first/req/get/`,
    type: 'get',
    data: {
      a: 100, b: 200, c: 300
    },
    success: function(result) {
      console.log(result);
    }
  });
</script>
```

■ AJAX

● jQuery (POST - Form Data)

```
<script src='http://code.jquery.com/jquery-3.1.1.min.js'></script>
<script>
  $.ajax({
    url: `/first/req/post/`,
    type: 'post',
    data: {
      a: 100, b: 200, c: 300
    },
    success: function(result) {
      console.log(result);
    }
  });
</script>
```

■ AJAX

● jQuery (POST - JSON Data)

```
<script>
$.ajax({
  url: `/first/req/json/`,
  type: 'post',
  data: JSON.stringify({
    a: 100, b: 200, c: 300
  }),
  headers: {
    'Content-Type': 'application/json'
  },
  success: function(result) {
    console.log(result);
  }
});
</script>
```

■ AJAX

● Axios (GET)

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
  const data = axios({
    url: '/first/req/get/',
    method: 'get',
    params: {
      'a': 10, 'b': 20, 'c': 30
    }
  });
  data.then(function (result) {
    console.log(result.data);
  });
</script>
```

■ AJAX

● Axios (POST - Form Data)

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
  const formData = new FormData();
  formData.append('a', 10);
  formData.append('b', 20);
  formData.append('c', 30);

  const data = axios({
    url: '/first/req/post/',
    method: 'post',
    data: formData
  });
  data.then(function (result) {
    console.log(result.data);
  });
</script>
```


■ AJAX

● Axios (POST - JSON Data)

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<script>
  const data = {
    'a': 10, 'b': 20, 'c': 30
  };

  const ajax = axios({
    url: '/first/req/json/',
    method: 'post',
    data: JSON.stringify(data),
    headers: {
      'Content-Type': 'application/json'
    }
  });

  ajax.then(function (result) {
    console.log(result.data);
  });
</script>
```

■ AJAX

- GET : ? 기호 다음 name=value 형식으로 입력 (Query Parameter)

- POST

- Form Data : FormData 객체 생성 후 데이터 입력

- JSON Data (Content-Type : application/json)

- JSON 문자열 데이터 입력

- 1) FormData → JSON 문자열

```
JSON.stringify(Object.fromEntries(formData))
```

- 2) Object → JSON 문자열

```
JSON.stringify(data)
```