

PHP POST ve GET Global Variable'ların Java ve CSharp'da Denkleri

İçindekiler

- A) Java Spring Framework'te \$_GET ve \$_POST Denkleri
- B) C# ASP.NET Framework'te \$_GET ve \$_POST Denkleri
- C) Notlar

A) Java Spring Framework'te \$_GET ve \$_POST Denkleri

i) GET

// Fetch URI ? Parameters

```
// @RequestMapping annotation'ı GET request  
// alan controller metodunu tanımlar.
```

```
// @RequestParam annotation'ı http isteğindeki q  
// parametresini metodun query parametresine bağlar.
```

```
@RequestMapping(value = "/search", method = RequestMethod.GET)  
public String search(@RequestParam("q") String query, Model model) {
```

```
    // Process search query  
    return "search-results";
```

```
}
```

veya

```
// @GetMapping annotation'ı GET request  
// alan controller metodunu tanımlar.
```

```
// @RequestParam annotation'ı http isteğindeki q  
// parametresini metodun query parametresine bağlar.
```

```
@GetMapping("/search")  
public String search(@RequestParam("q") String query, Model model) {
```

```
    // Process search query  
    return "search-results";
```

```
}
```

```
// Not: @RequestMapping her tür http metod türünü kapsar.
```

```
// @RequestMapping'in GET için kısaltılmış hali @GetMapping'tir.
```

ii) GET - 2

// Fetch URI Path Parameters

```
// @RequestMapping annotation'ı GET request  
// alan controller metodunu tanımlar.
```

```
// @PathVariable annotation'ı http isteğindeki url'de  
// id bölümünde yer alan string'i metodun userID  
// parametresine bağlar.
```

```
@RequestMapping(value = "/users/{id}", method = RequestMethod.GET)  
public String getUser(@PathVariable("id") int userId, Model model) {
```

```
    // Retrieve user by ID  
    return "user-details";
```

```
}
```

veya

```
// @GetMapping annotation'ı GET request  
// alan controller metodunu tanımlar.
```

```
// @PathVariable annotation'ı http isteğindeki url'de  
// id bölümünde yer alan string'i metodun userID  
// parametresine bağlar.
```

```
@GetMapping("/users/{id}")  
public String getUser(@PathVariable("id") int userId, Model model) {
```

```
    // Retrieve user by ID  
    return "user-details";
```

```
}
```

```
// Not: @RequestMapping her tür http metod türünü kapsar.
```

```
// @RequestMapping'in GET için kısaltılmış hali @GetMapping'tir.
```

iii) POST

// Fetch Http Request Body
// Parameters Separately

```
// @RequestMapping annotation'ı POST request  
// alan controller metodunu tanımlar.
```

```
// @RequestParam annotation'ı http isteğinin gövde-  
// sindeki name parametresini metodun name paramet-  
// resine bağlar, http isteğinin gövdesindeki email pa-  
// rametresini metodun email parametresine bağlar.
```

```
@RequestMapping(value = "/submitForm", method = RequestMethod.POST)  
public String submitForm(@RequestParam("name") String name,  
                        @RequestParam("email") String email,  
                        Model model) {  
  
    // Process form data  
    return "success";  
}
```

veya

```
// @PostMapping annotation'ı POST request  
// alan controller metodunu tanımlar.
```

```
// @RequestParam annotation'ı http isteğinin gövde-  
// sindeki name parametresini metodun name paramet-  
// resine bağlar, http isteğinin gövdesindeki email pa-  
// rametresini metodun email parametresine bağlar.
```

```
@PostMapping("/submitForm")  
public String submitForm(@RequestParam("name") String name,  
                        @RequestParam("email") String email,  
                        Model model) {  
  
    // Process form data  
    return "success";  
}
```

```
// Not: @RequestMapping her tür http metod türünü kapsar.  
//      @RequestMapping'in POST için kısaltılmış hali @PostMapping'tir.
```

iv) POST - 2

// Fetch Http Request Body
// Content Entirely

// @RequestMapping annotation'ı POST request
// alan controller metodunu tanımlar.

// @RequestBody annotation'ı http isteğinin gövde-
// sindeki tüm içeriği metodun data parametresine
// bağlar.

// Örn; JSON veya XML veya Binary veriler bu
// şekilde çekilebilir.

@RequestMapping(value = "/submitForm", method = RequestMethod.POST)
public String submitForm(@RequestBody ExampleData data) {

 // Process form data
 return "example";

}

veya

// @PostMapping annotation'ı POST request
// alan controller metodunu tanımlar.

// @RequestBody annotation'ı http isteğinin gövde-
// sindeki tüm içeriği metodun data parametresine
// bağlar.

// Örn; JSON veya XML veya Binary veriler bu
// şekilde çekilebilir.

@PostMapping("/submitForm")
public String submitForm(@RequestBody ExampleData data) {

 // Process form data
 return "example";

}

B) C# ASP.NET Framework'te \$_GET ve \$_POST Denkleri

i) GET

// Fetch URI ? Parameters

```
// [HttpGet] attribute'u metodun GET  
// request'i handle edeceğini tanımlar.
```

```
// [Route(...)] attribute'u hangi URL'den  
// bu controller metodun tetikleneceğini  
// belirtir.
```

```
[HttpGet]  
[Route("api/users/{id}")]  
public IActionResult GetUser(int id)  
{  
  
    // code to retrieve user with the specified ID  
    return Ok(user);  
  
}
```

ii) POST

```
// [HttpPost] attribute'u metodun POST  
// request'i handle edeceğini tanımlar.
```

```
// [Route(...)] attribute'u hangi URL'den  
// bu controller metodun tetikleneceğini  
// belirtir.
```

```
[HttpPost]  
[Route("api/users")]  
public IActionResult CreateUser([FromBody] UserModel model)  
{  
  
    // code to create a new user based on the data in the model  
    return CreatedAtAction(nameof(GetUser), new { id = newUser.Id }, newUser);  
  
}
```

C) Notlar

- C'de nasıl int main(argv) fonksiyon parametresi konsoldan gelen argümanları alıyor aynı onun gibi Java'da da http isteğinden gelen (@RequestParam annotation'ındaki) argümanlar metod parametresine argüman olarak atanıyor.

- Java'da controller'daki metodlar üzerlerine annotation olarak get ve post gibi request'leri handle ederler, C#'da controller'daki metodlar üzerlerine attribute olarak get ve post gibi request'leri handle ederler.

- Java'da @GetMapping ve @PostMapping gelen istekleri controller metoduna bağlar. Controller metoduna Get eşleme veya Post eşleme yapar.

- C#'daki controller'daki metotlar üzerinde Route() attribute'u ile hangi url'den geldiğinde ilgili metot tetiklensin tanımı girildiği gibi diğer mvc yapısındaki programlama dillerinde de controller'daki metotların üzerine route tarzı ifadeler geldiğinde aynı anlamı taşır. Örneğin Python'da Django framework'ünde şöyle;

```
@app.route('/select/tables_data', methods=['POST'])
def select_data():

    try:
        ...

    except Exception as e:
        ...
```

Kaynaklar

ChatGPT:

- what is the equivalent of \$_POST variable in spring

- what is the equivalent of \$_GET variable in spring

- what is the equivalent of \$_POST variable in asp.net c#

- what is the difference between @RequestParam and @RequestBody annotations in Java Spring Framework?

- What is the difference between @RequestMapping, @GetMapping and @PostMapping annotations in java spring framework?

- what is the equivalent of \$_POST variable in asp.net c#

- How to use [HttpGet] in c#

- How to use [HttpPost] in c#

<https://stackoverflow.com/questions/13715811/requestparam-vs-pathvariable>

<https://stackoverflow.com/questions/11291933/requestbody-and-responsebody-annotations-in-spring>

<https://stackabuse.com/get-http-post-body-in-spring/>

<https://www.codesenior.com/tutorial/Spring-RequestMapping-PathVariable-RequestBody-Ve-ResponseBody-Annotasyonlari>

<https://bushansirgur.in/spring-mvc-valid-annotation-with-example/>