



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Spring Workshop



3 - Web

Part 1: Basics



Agenda

- Common Annotations & Classes
- DTOs & Records
- BeanValidation
- WebTests



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Common Annotations (already using)

- @RestController
- @RequestMapping
 - URL
 - Shortcuts for different Http Methods
- @PathVariable
 - Used for Url Matching



Common Annotations (new)

- `@RequestBody`
 - Maps the complete body into an object
- `@RequestPart`
 - Multipart Form
 - e.g. file upload
 - also support type conversion
- `@RequestParam`
 - Query Strings
 - Form Data



Common Annotations Classes (new)

- **ResponseEntity**
 - Allows to set headers and status
 - e.g. Content-Disposition for download
 - superior @ResponseBody (included in @RestController)
- **HttpEntity**
 - Like ResponseEntity but for @RequestBoy



BeanValidation

- Standard with built-in validators
- <https://jakarta.ee/specifications/bean-validation/3.0/apidocs/>
- Examples
 - @NotNull
 - @NotBlank
 - @Min
 - @Past
- @Valid: Recursive Application



BeanValidation

- Standard with built-in validators
- <https://jakarta.ee/specifications/bean-validation/3.0/apidocs/>
- Examples
 - @NotNull
 - @NotBlank
 - @Min
 - @Past
 - @Pattern
- @Valid: Recursive Application



BeanValidation Example 1/2

```
public class Customer {  
  
    private Long id;  
    private String firstname;  
  
    @NotEmpty  
    private String lastname;  
  
    private Boolean signedGdpr;  
}
```



BeanValidation Example 2/2

```
public class CustomersController {  
    @PostMapping  
    public void add(@RequestBody @Valid Customer customer) {  
        this.repository.add(customer);  
    }  
}
```



DTOs

- Data Transfer Object
- Heavily used between
 - Modul Boundaries
 - Endpoints
- Can be constructed via record
- Support BeanValidation



DTOs

```
public record AddCustomer(  
    @NotBlank String firstname,  
    @NotBlank String lastname  
) {}  
  
public class CustomersRepository {  
    public Instant add(AddCustomer addCustomer) {  
        var customer = new Customer(  
            this.id++,  
            addCustomer.firstname(),  
            addCustomer.lastname(),  
            showGdpr  
        );  
        customers.add(customer);  
        return LocalDate.now().atStartOfDay().toInstant(ZoneOffset.UTC);  
    }  
}
```



DTOs

Validate on Controllers



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

WebTests with WebTestClient

- @AutoConfigureMockMvc starts server
- Allows us to send HTTP Request to Controllers
- WebTestClient
 - more powerful than MockMvc
 - Fluent
 - JSONPath for assertions: <https://jsonpath.com/>



Example WebTest

```
@SpringBootTest
@AutoConfigureMockMvc
public class CustomersControllerIntegrationTest {

    @Autowired
    WebTestClient webTestClient;

    @Test
    void testRepo() {
        webTestClient.get().uri("/api/customers").exchange().expectStatus().isOk();
    }
}
```



Example WebTest

```
public class CustomersControllerIntegrationTest {  
    @Test  
    void testAddCustomer() {  
        AddCustomer addCustomer = new AddCustomer("Franz", "Maier");  
        webTestClient  
            .post()  
            .uri("/api/customers")  
            .bodyValue(addCustomer)  
            .exchange()  
            .expectStatus()  
            .isOk();  
  
        webTestClient  
            .get()  
            .uri("/api/customers")  
            .exchange()  
            .expectBody()  
            .jsonPath("[0].firstname")  
            .isEqualTo("Franz");  
    }  
}
```



Example WebTest

```
@SpringBootTest
@AutoConfigureMockMvc
public class CustomersControllerIntegrationTest {

    @Test
    void testAddInvalidCustomer() {
        AddCustomer addCustomer = new AddCustomer("Franz", "");
        webTestClient
            .post()
            .uri("/api/customers")
            .bodyValue(addCustomer)
            .exchange()
            .expectStatus()
            .isBadRequest();
    }
}
```



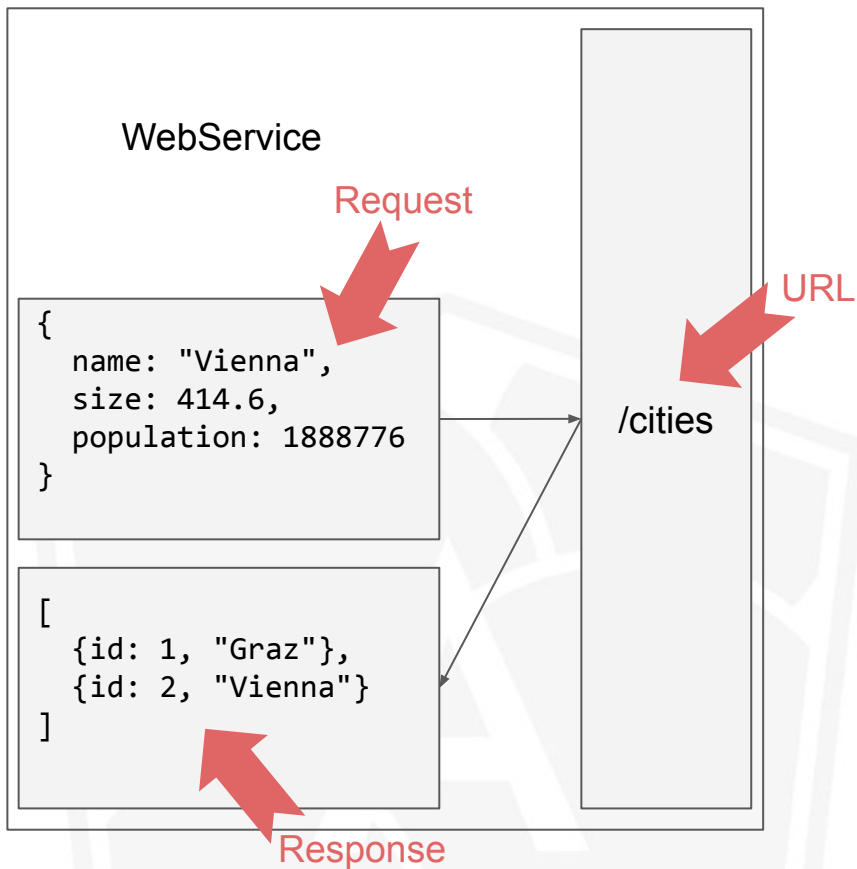
Part 2: OpenAPI



Agenda

- Swagger / OpenAPI
- UI
- Code Generator
- ShowCase on Customers



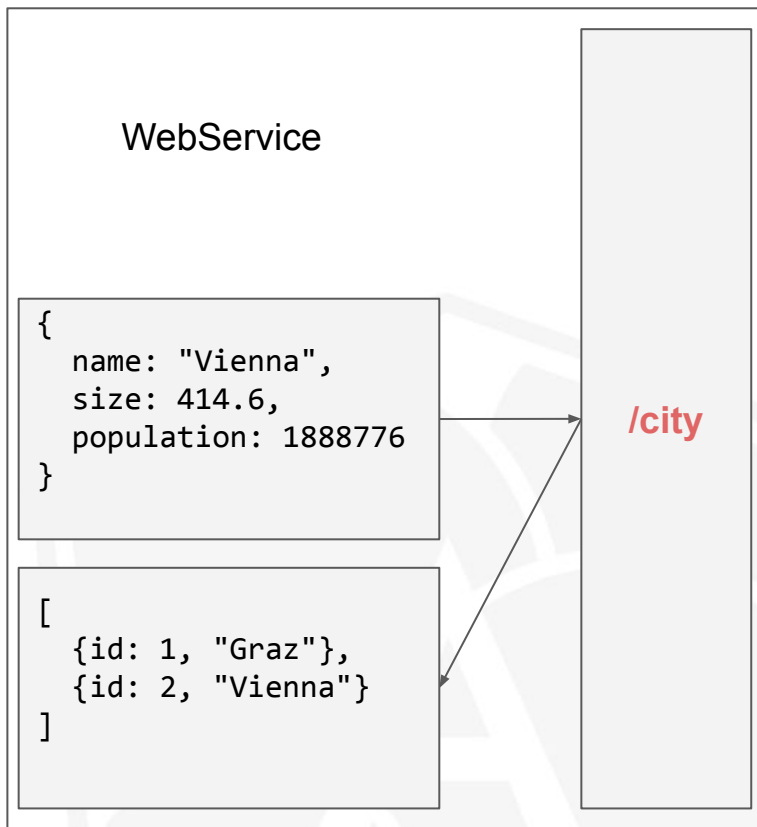


Client / Javascript

```
function addCity(name, size, population) {  
  return fetch(  
    '/cities',  
    {name, size, population}  
  );  
}
```



Out-of-Sync Problems

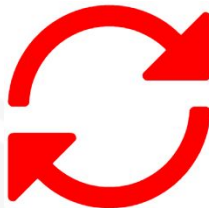
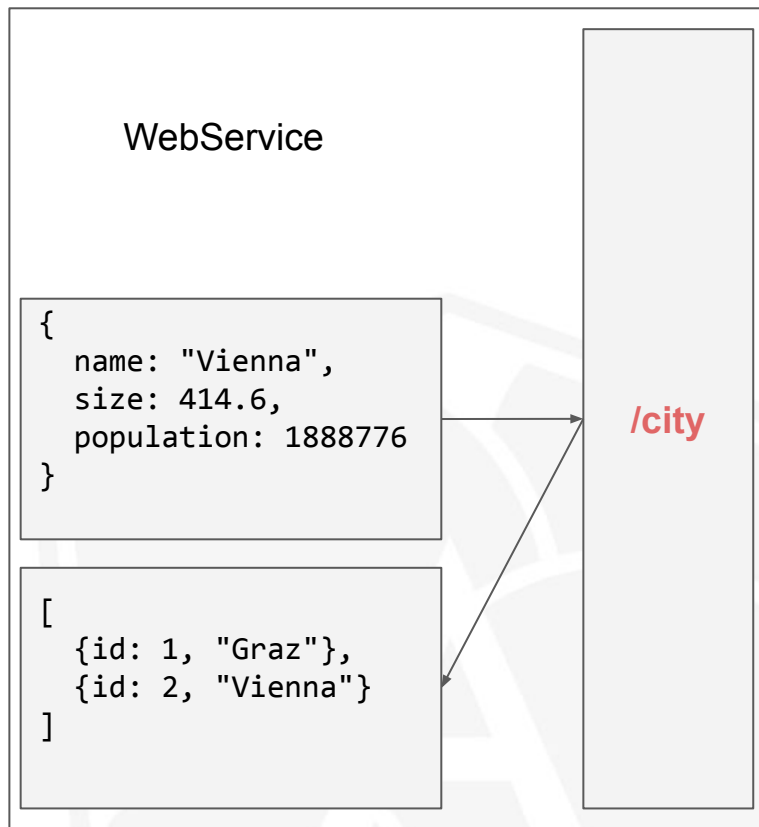


Client / Javascript

```
function addCity(name, size, population) {  
  return fetch(  
    '/cities',  
    {name, size, population}  
  );  
}
```



Out-of-Sync Problems

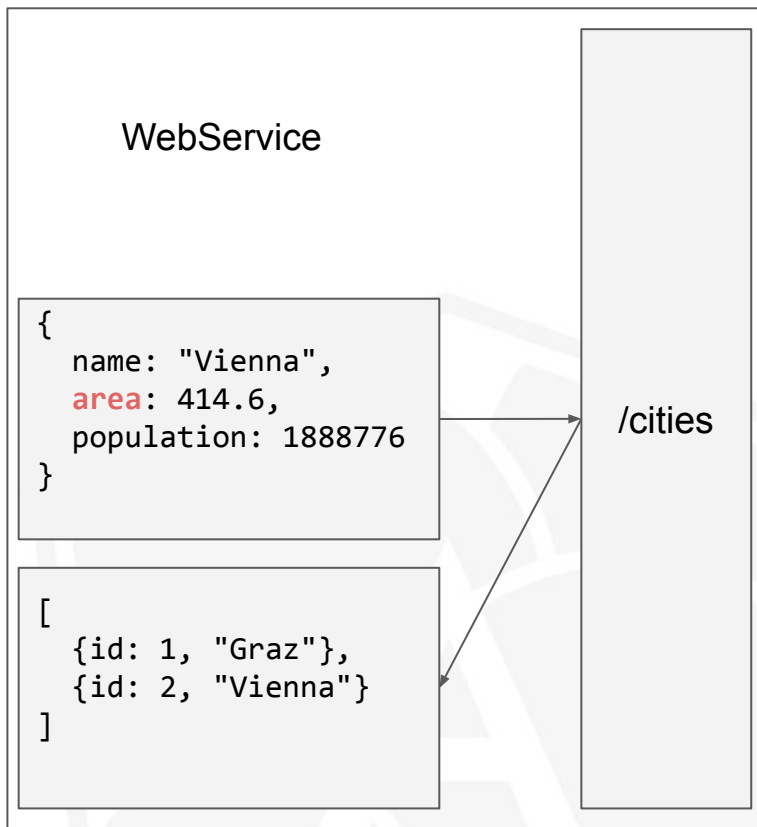


Client / Javascript

```
function addCity(name, size, population) {
  return fetch(
    "/cities",
    {name, size, population}
  );
}
```



Out-of-Sync Problems



Client / Javascript

```
function addCity(name, size, population) {  
  return fetch(  
    '/cities',  
    {name, size, population}  
  );  
}
```



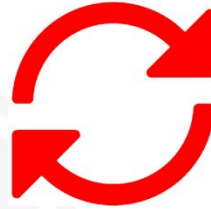
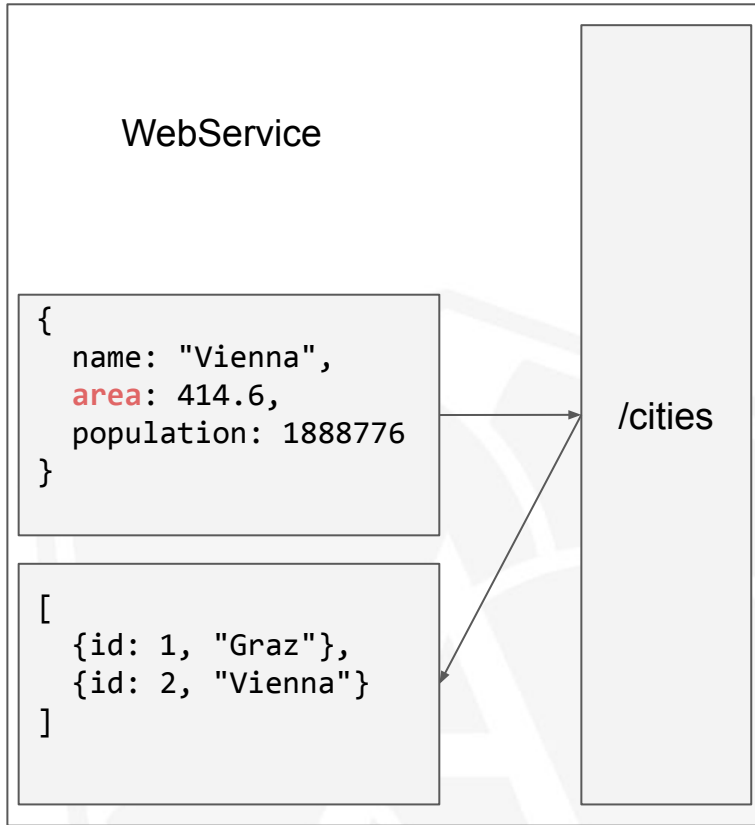
Out-of-Sync Problems



Client / Javascript

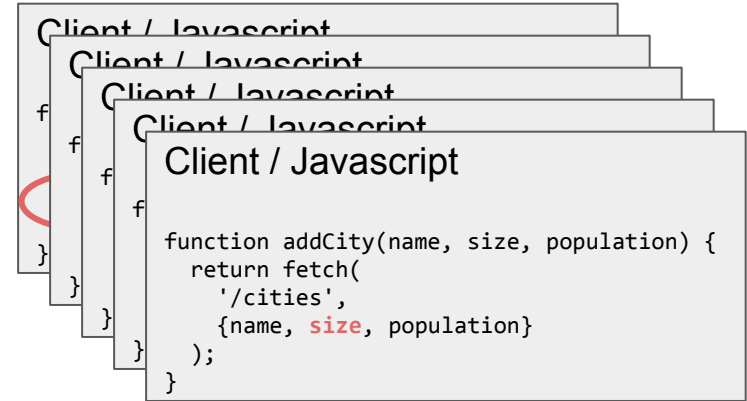
```
function addCity(name, size, population) {
  return fetch(
    '/cities',
    {name, size, population}
  );
}
```





Out-of-Sync Problems

(Microservices Version)



Ruby Client

Python Client

C# Client

PHP Client

Java Client

Objective-C Client



ARCHITECTS
INSIDE KNOWLEDGE

OpenAPI

Solution for developing, documenting and maintaining Web APIs

- Elements
 - URL Paths
 - Response Types
 - Request Types
- Code duplication
- On-Going synchronisation efforts
- Resulting bugs



The Ecosystem

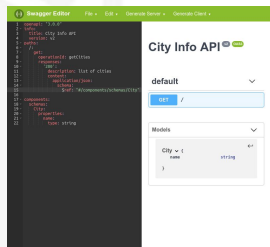
Documentation



Specification
v2 = Swagger Spec.
v3 = OpenAPI



Tools



Editor

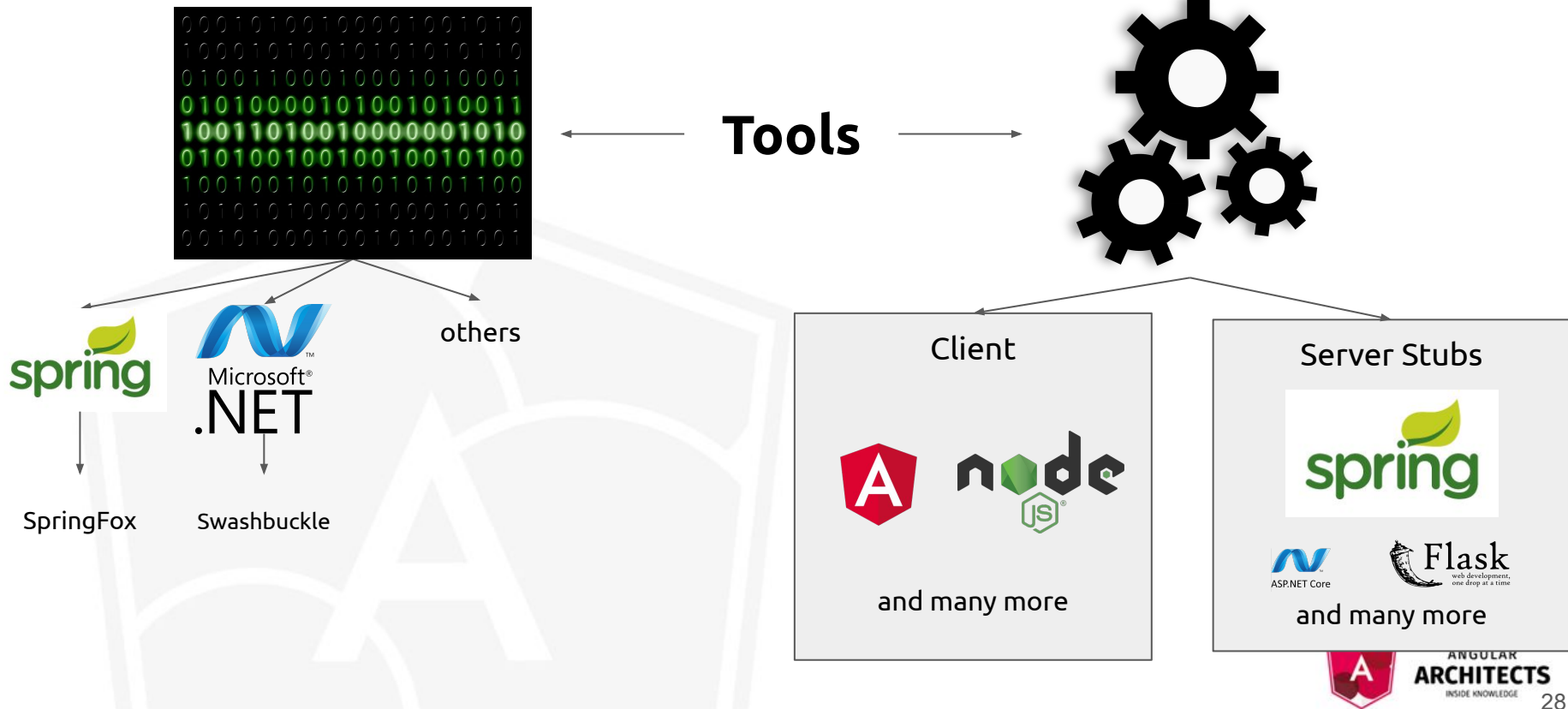


ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

Code First

API First
(Code Generators)

Tools



Dependencies

- SpringDoc
 - org.springdoc:springdoc-openapi-starter-webmvc-ui:2.0.0-RC1
 - Success of SpringFox
 - Code Generator Plugins for Gradle
 - Lots of Customisations
- openapi-generator-cli
 - NPM Package
 - Triggers Java



Important Annotations

- @Tag
 - Defines Name of generated Angular Service
- @Operation
 - Defines Name of Method



ANGULAR
ARCHITECTS
INSIDE KNOWLEDGE

OpenAPI definition v0 OAS3

/v3/api-docs

Servers

<http://localhost:8080> - Generated server url

Holidays

[GET](#) /api/holidays

[PUT](#) /api/holidays

[POST](#) /api/holidays

[GET](#) /api/holidays/{id}

[DELETE](#) /api/holidays/{id}

[GET](#) /api/holidays/{id}/cover

Parameters

[Try it out](#)

| Name | Description |
|----------------------------|-------------|
| id <small>required</small> | id |

integer(int64)
(path)

Responses

| Code | Description | Links |
|------|-------------|----------|
| 200 | OK | No links |

OK



Part 3: Upload & Download



Controller for File Uploads

```
@RequestMapping("/api/customers")
@RestController
@Tag(name = "Customers")
public class CustomersController {

    @PostMapping(consumes = MediaType.MULTIPART_FORM_DATA_VALUE)
    @Operation(operationId = "add")
    public boolean add(
        @RequestPart AddCustomer addCustomer,
        @RequestPart MultipartFile profile
    ) throws IOException {

        var filename = profile.getOriginalFilename();
        var path = Path.of("", "filestore", filename);
        this.repository.add(addCustomer, filename);
        cover.transferTo(Path.of("", "filestore", filename));
        return true;
    }
}
```



Angular Code for Upload

```
/* Legacy Code without OpenAPI */  
const { name, description } = holiday;  
if (holiday.cover) {  
  formData.append('cover', holiday.cover);  
}  
  
const blob = new Blob([JSON.stringify({ name, description })], {  
  type: 'application/json',  
});  
  
formData.append('holidayDto', blob);  
assertDefined(holiday.cover);  
  
const saveRequest: SaveRequest = {  
  cover: holiday.cover,  
  holidayDto: blob as HolidayDto,  
};  
  
this.#httpClient.post('http://localhost:8080/api/holidays', formData);
```

