4.3 Implementation

4.3.2 Server

4.3.2.1 Basic Architecture

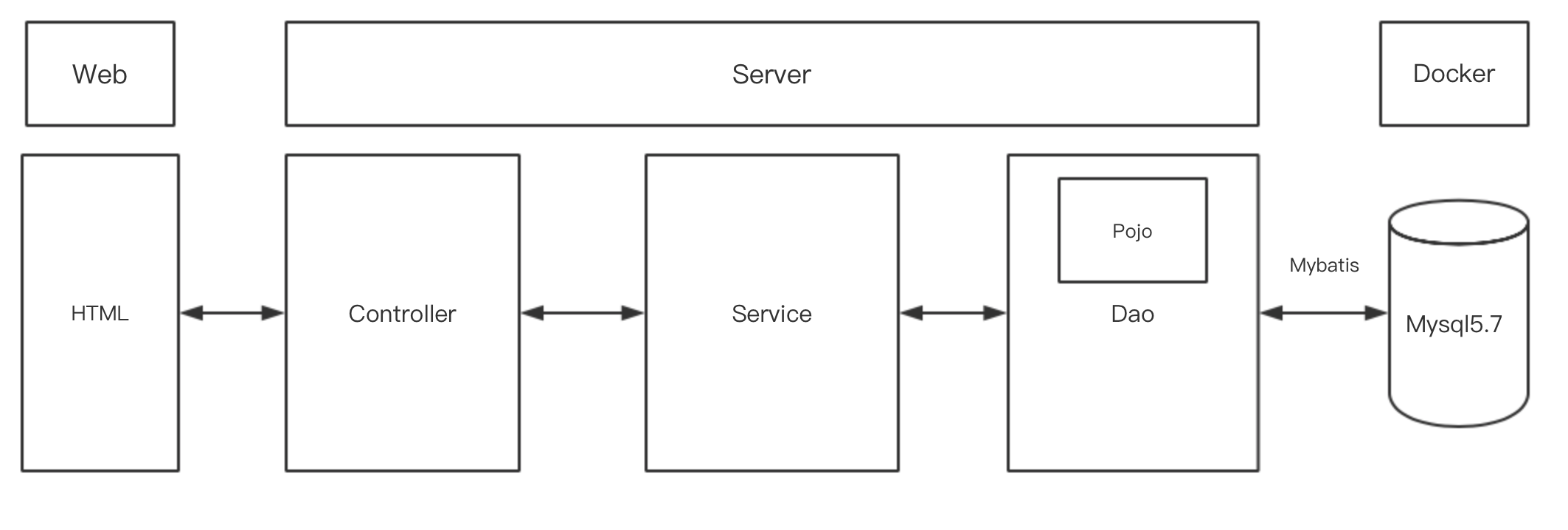
Running Environment:Linux CentOS 7

Frameworks and tools:

1. Springboot: a stand-alone Spring applications using embed Tomcat.
2. Mybatis: a first class persistence framework with support for custom SQL, stored procedures and advanced mappings.
3. Mysql:a classic database.
4. Docker: a easiest and fastest tool to use containers(mainly for Mysql5.7 in the project).

Layers:

1. Pojo: definition of Java Object(History,Collection,Methods),automatic generated by Mybatis-generator.
2. Dao: Execute retrieve or insert operations in the database(base on Object defined in Pojo Layer),automatic generated by Mybatis-generator.
3. Service: Implement core logic in project and process requests from Controller Layer.
4. Controller: Responding all kinds of web requests.



**Figure 4.x** Basic Architecture

4.3.2.2 Database Design

The History table design, Collection table design and Method table design in database are shown in Table 4.1, Table 4.2 and Table 4.3, respectively.

**Table 4.1** History Table Design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **history** | | | |
| **Attribute** | **Type** | **Null/Non-Null** | **constraint condition** | **Description** |
| **id** | int(11) | Non-Null | Primary key | The id of history  auto-increment |
| **method** | varchar(10) | Non-Null |  | method of request |
| **request** | varchar(200) | Non-Null |  | The URL of request |
| **body** | json | Null |  | content of response |
| **cookie** | json | Null |  | Cookie of response from server |
| **header** | json | Null |  | header of response from server |
| **date** | datetime | Null |  | The date of request |
| **time** | int(11) | Null |  | The time of response |

**Table 4.2** Collection Table Design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **collection** | | | |
| **Attribute** | **Type** | **Null/Non-Null** | **constraint condition** | **Description** |
| **id** | int(11) | Non-Null | Primary key | The id of collection  auto-increment |
| **method** | varchar(10) | Non-Null |  | method of request |
| **request** | varchar(200) | Non-Null |  | The URL of request |

**Table 4.3** Method Table Design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **method** | | | |
| **Attribute** | **Type** | **Null/Non-Null** | **constraint condition** | **Description** |
| **id** | int(11) | Non-Null | Primary key | The id of method |
| **method** | varchar(10) | Non-Null |  | method of request |

4.3.2.3 Interface Design

**Here is the basic design for interfaces in the project:**

//Get HTTP history list

@GetMapping("/GetHistory")

//Get HTTP history details by id  
@GetMapping("/GetHistoryById/{id}")

//Insert HTTP history  
@PostMapping("/InsertHistory/{request}/{method}/{body}/{cookie}/{header}/{time}")

//Insert HTTP history by JSON  
@PostMapping(value = "/InsertHistoryByJSON",produces = "application/json;charset=UTF-8")

//Delete HTTP history by id  
@DeleteMapping("/DeleteHistoryById/{id}")

//Get all the HTTP methods  
@GetMapping("/GetMethods")

//Get HTTP collection list  
@GetMapping("/GetCollections")

//Add HTTP collection by id  
@PostMapping("/AddToCollection/{id}")

//Delete HTTP collection by id  
@DeleteMapping("DeleteCollectionById/{id}")

//Analyze URL request by method and URL address  
@GetMapping("/AnalyseURL/{method}/{url}")