Bereitstellen und Konsumieren von REST APIs mit Spring

Agenda

01 REST in der Theorie

02 HTTP

03 Demo Bereitstellen einer REST API

04 Demo Konsumieren einer REST API

05 Fazit

01

REST in der Theorie

Representational State Transfer

Es handelt sich um einen Architekturstil

Kommunikation über HTTP

Klar definierte Zugangspunkte, welche über URIs aufgerufen werden

REST folgt sechs Prinzipien

01

Sechs Prinzipien von REST

Client-Server-Prinzip

- Klare Trennung von Client und Server
  (Bsp. Frontend – Backend)
- Möglichkeit der unabhängigen Weiterentwicklung der beiden Komponenten

Zustandslosigkeit

- Keine Zwischenspeicherung von Daten des Clients
- Verschiedene Anfragen werden als voneinander unabhängige Transaktionen behandelt

Caching

- Bei allen gesendeten Dateien wird explizit angegeben, ob diese zwischengespeichert werden können

Einheitliche Schnittstelle

- Prinzip der Allgemeingültigkeit
- Identische Requests geben jeder Zeit das exakt gleiche Ergebnis, mit unterschiedlichen Darstellungsformen (Json. XML) zurück

Mehrschichtige Systeme

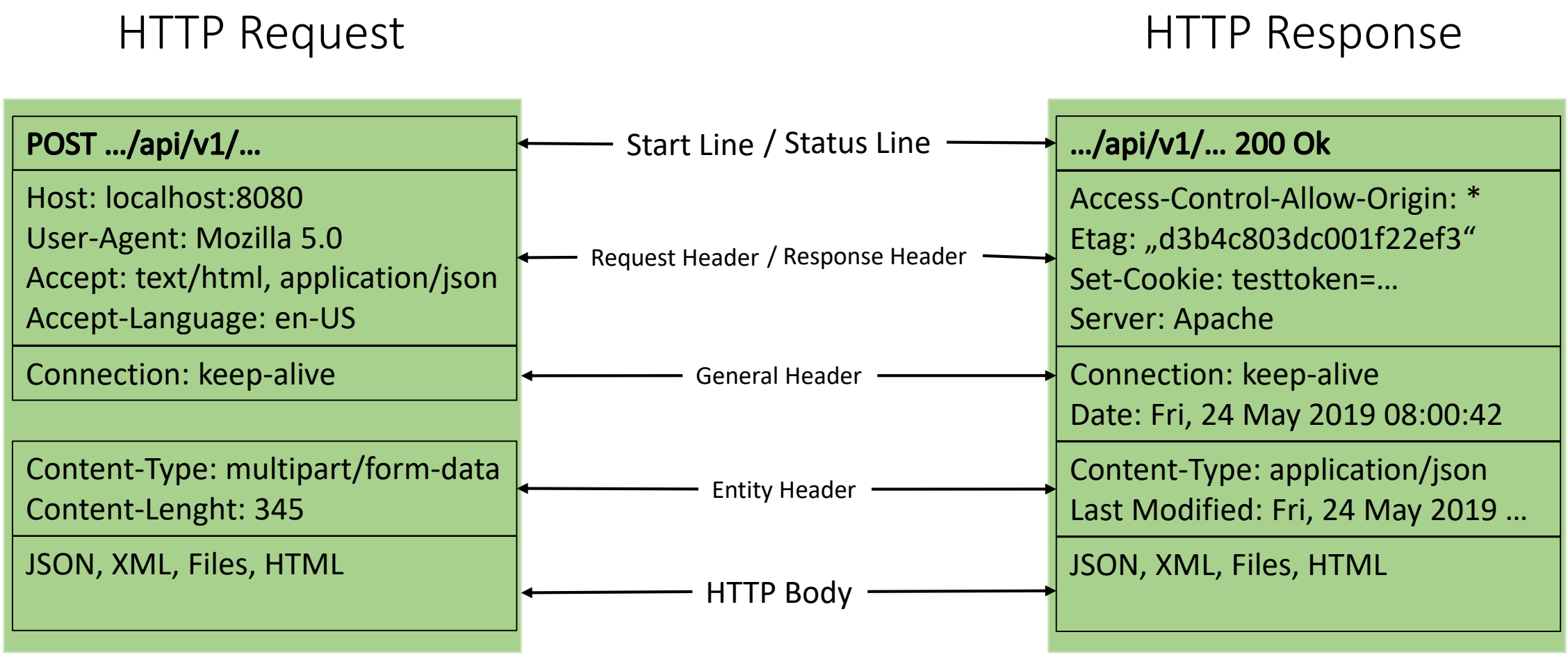
- Hierarchisches Schichtenmodell
- In sich geschlossene Komponenten
- Dem Client wird nur die Schnittstelle angeboten, alle darunterliegenden Schichten bleiben verborgen

Code-On-Demand

- Optional
- Dem Client wird erlaubt Quellcode herunterzuladen und zu modifizieren

02

HTTP Grundlagen



HTTP Grundlagen

Statuscode	Bedeutung
1xx	Informationen
2xx	Erfolgreiche Operation
3xx	Umleitung
4xx	Client Fehler
5xx	Server Fehler

HTTP Grundlagen

Statuscode	Nachricht	Bedeutung
200	Ok	Die Anfrage wurde erfolgreich bearbeitet
201	Created	Das gesendete Objekt wurde erfolgreich erstellt. Meist wird zusätzlich eine Url, unter der das neue Objekt erreichbar ist, mitgeschickt
204	No Content	Die Anfrage wurde erfolgreich bearbeitet, jedoch enthält die Antwort keine Daten. Wird z.B oft bei DELETE verwendet
400	Bad Request	Die Anfrage wurde nicht richtig aufgebaut und dem Server fehlen Daten
401	Unauthorized	Die Anfrage benötigt eine Authentifizierung
403	Forbidden	Der Client hat nicht die nötigen Rechte, um eine Operation durchzuführen
404	Not Found	Die angeforderte Ressource wurde nicht gefunden
500	Internal Server Error	Serverseitig kommt es zu einem Fehler und die Anfrage kann nicht richtig bearbeitet werden

03 Demo: Bereitstellen einer REST API mit Spring

Ziel

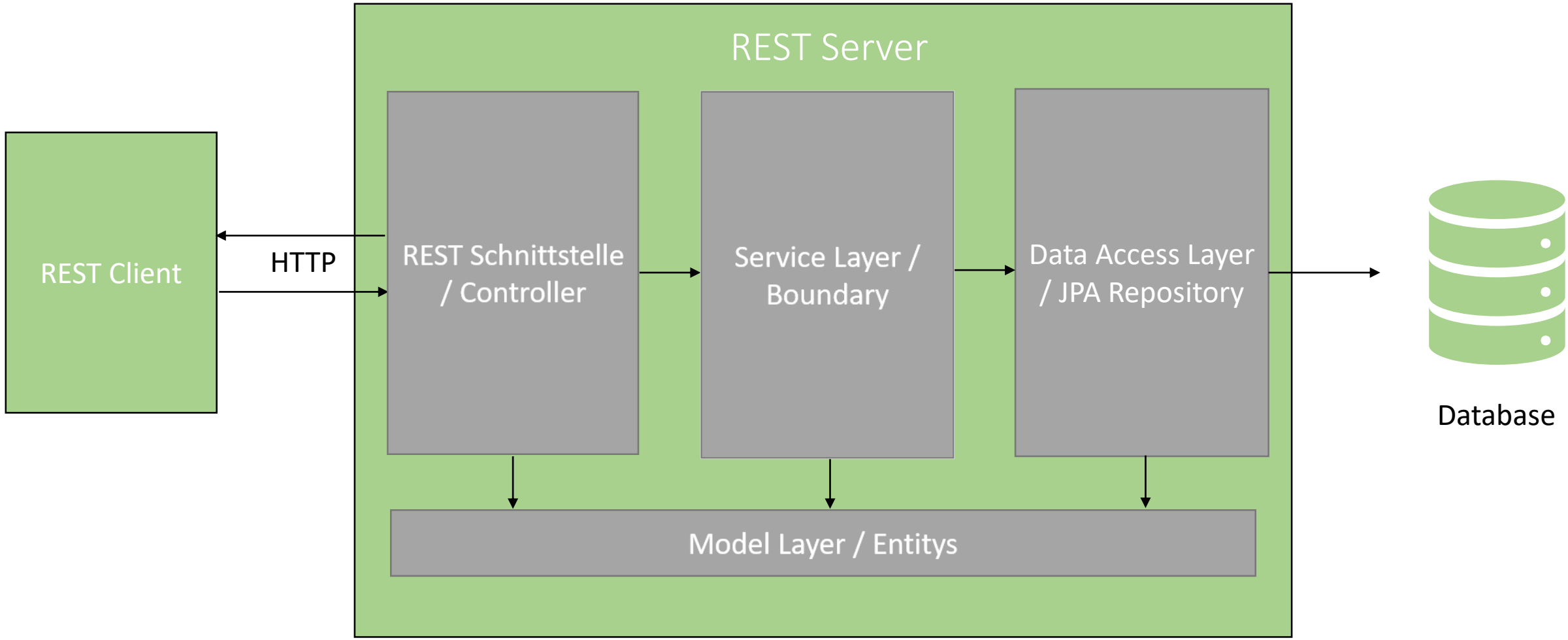
Entwickeln einer Client-Server-Applikation, welche es erlaubt ein Telefonbuch zu verwalten.

Telefonbucheinträge können
- Abgefragt werden
- Erstellt werden
- Geändert werden
- Gelöscht werden

Die Einträge werden auf einer lokalen MariaDB gespeichert

GitHub: https://github.com/IRuFFeYI/SpringRestApi

03 Demo: Bereitstellen einer REST API mit Spring



Model Layer / Entitys

PhonebookEntry
- id: int - prename: String - name : String - phoneNumber: String
+ getter-/ setter-Methoden

Service Layer / Boundary

PhonebookManager

Funktionen:

- Enthält die Geschäftslogik
- Validiert die Eingaben, welche der Client an den Server schickt
- Nutzt das Repository für den Datenbankzugriff

Rest Schnittstelle / Controller

PhonebookController

Erreichbar unter der Adresse:
…/api/v1/phonebookentrys

Funktionen:

- Einen Eintrag mit einer bestimmten ID zurückgeben
- Alle Einträge zurückgeben
- Einen Eintrag entgegennehmen und in die Datenbank speichern
- Einen bereits vorhandenen Eintrag ändern
- Einen Eintrag löschen
- Einträge nach Namen oder Vornamen filtern

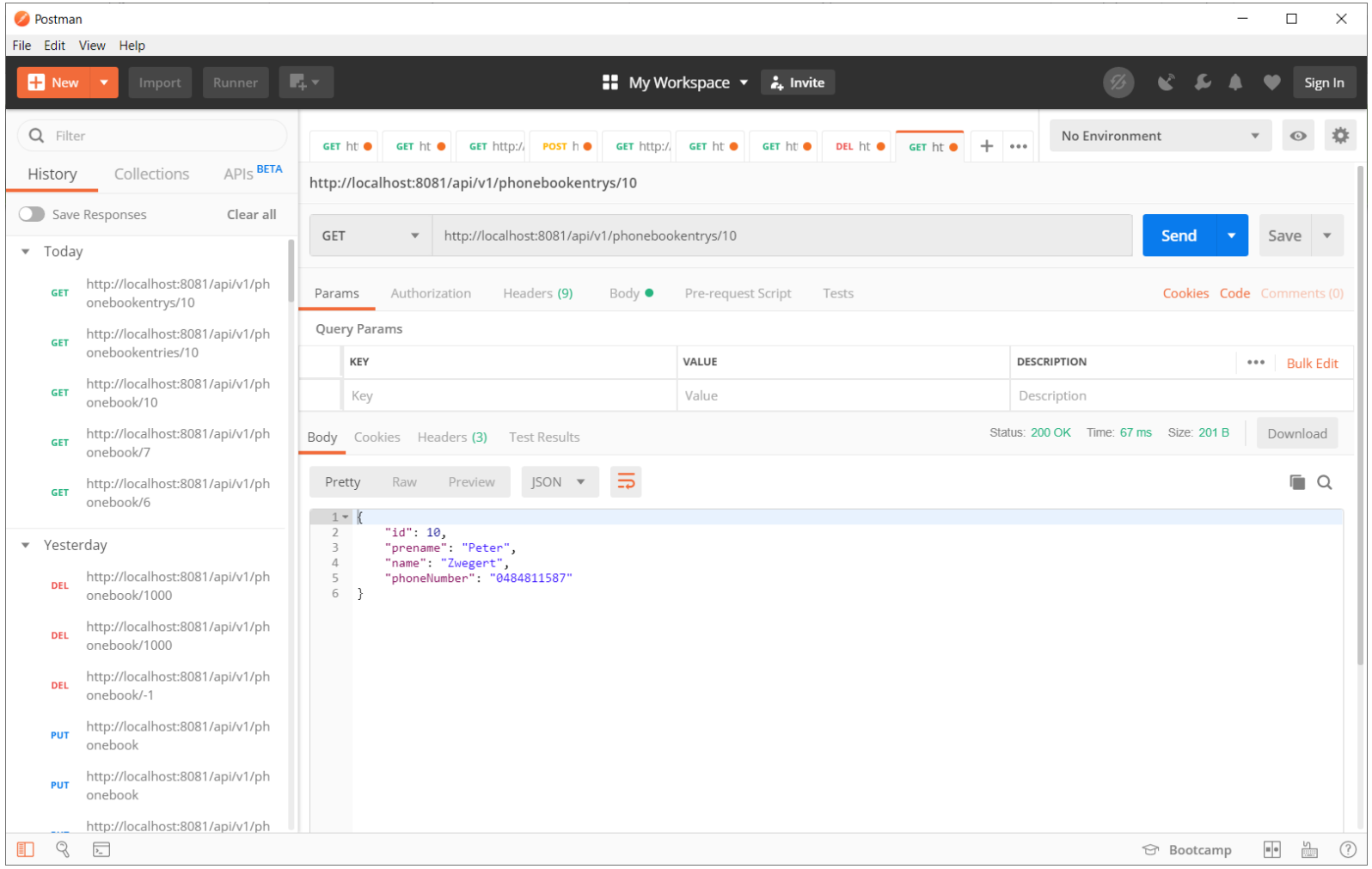
PhonebookController

boundary: PhonebookManager

+ getPhonebookEntryById(int id):
+ createPhonebookEntry(PhonebookEntry entry)
+ updatePhonebookEntry(PhonebookEntry entry)
+ deletePhonebookEntry(int id)
+ filterPhonebookEntries(String name, String prename)

Postman

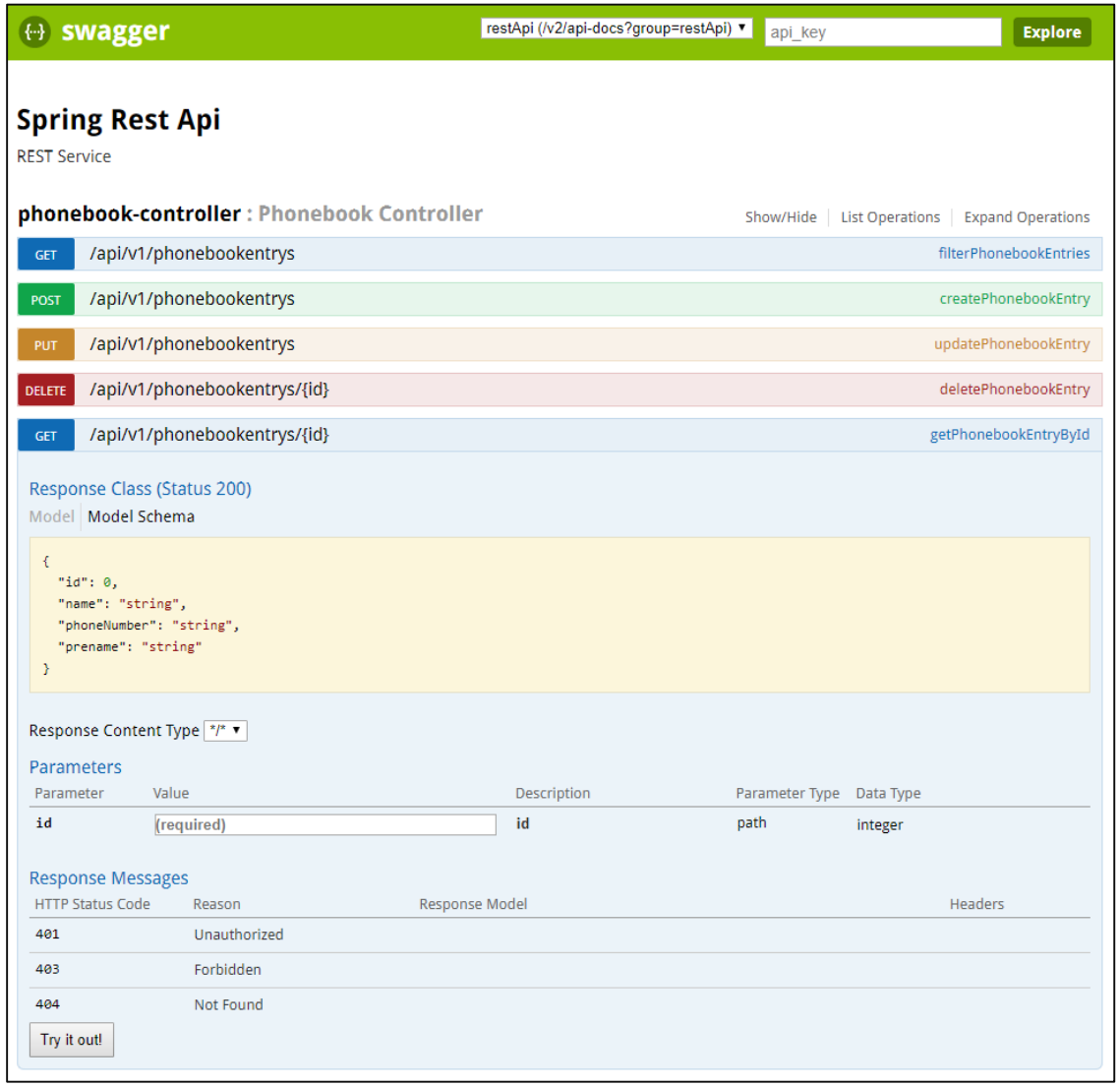
- Tool um HTTP-Requests aufzubauen und an einen Server zu schicken
- Empfangen von HTTP-Responses und Anzeigen des Bodies, des Statuscodes und weiteren Informationen



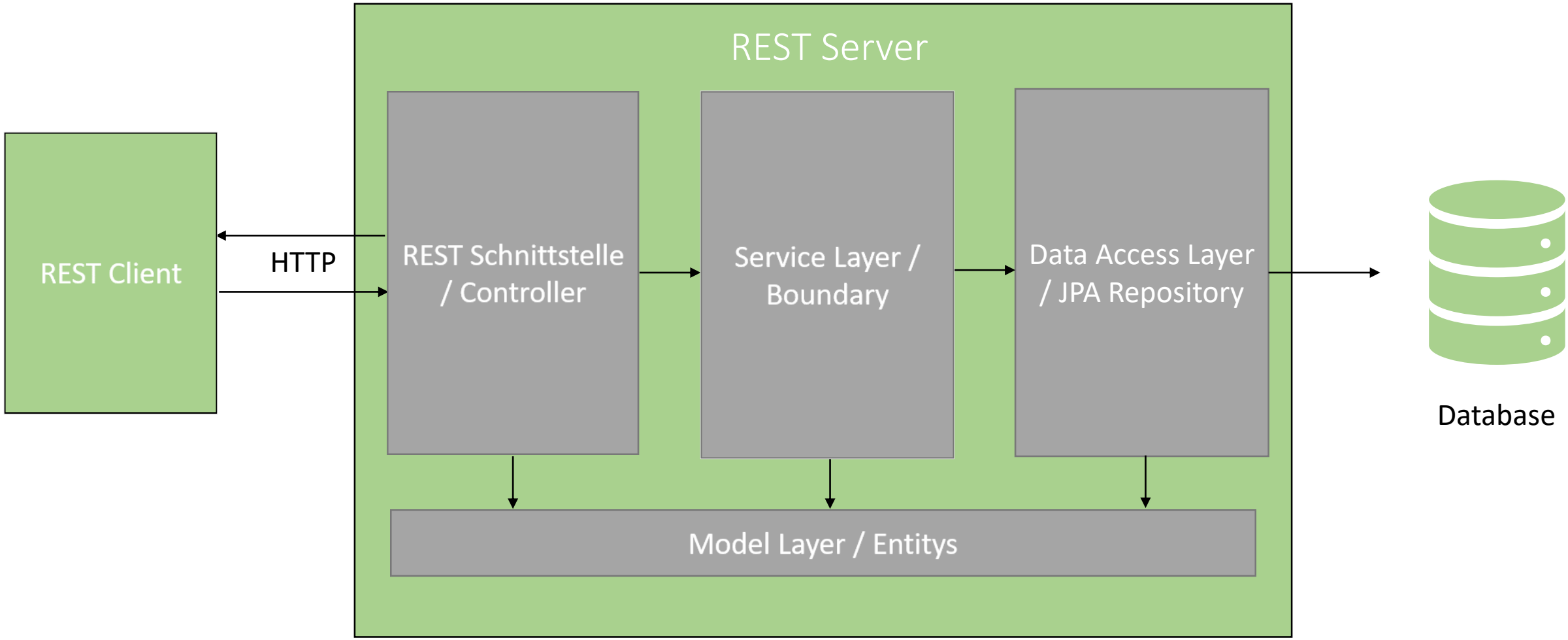
Swagger

- Swagger ist ein Open-Source-Framework
- Swagger bietet
  - Automatisierte Dokumentation
  - Code-Generierung
  - Testfallgenerierung

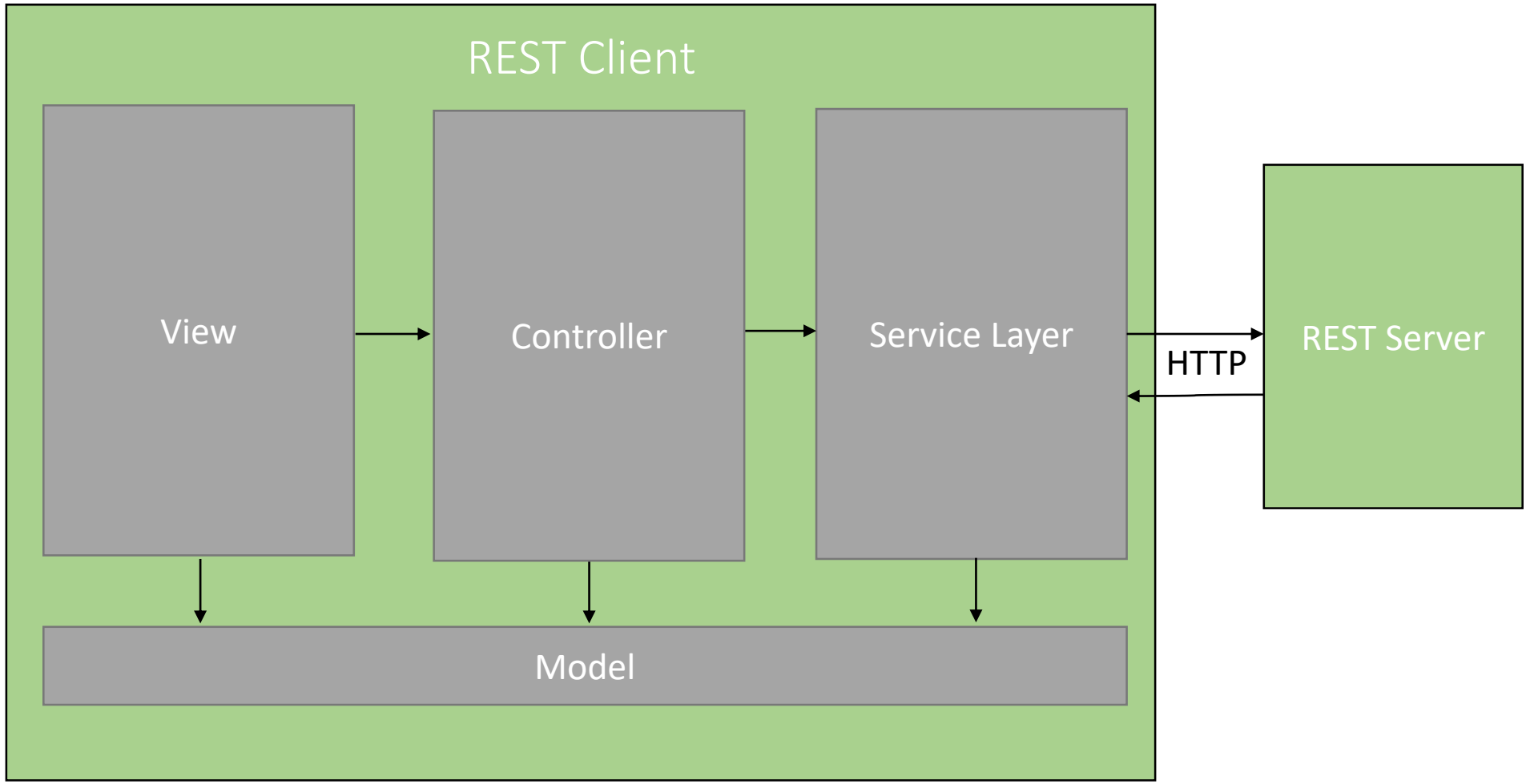
- Swagger-Ui um Schnittstellen aufzurufen



04 Demo: Konsumieren einer REST API mit Spring



04 Demo: Konsumieren einer REST API mit Spring



GitHub:
https://github.com/IRuFFeYI/ConsumeRestSpring

05

Fazit

+ Generelle Vorteile von Spring
  + Leichtgewichtig
  + Einfach zu implementieren
  + Klar verständlich
  + Projekt ist extrem schnell erstellt und einsatzbereit
+ Sehr gute Kompatibilität mit anderen Dependencies (Datenbankzugriff, Security)
- Oft werden in Beispielen, Dokumentationen oder Foreneinträgen Komponenten aus Spring und Spring Boot vermischt

Vielen Dank für Eure Aufmerksamkeit

Fragen?

Quellen

- bealdung. (26. 10 2018). Ant vs Maven vs Gradle. Von baeldung: https://www.baeldung.com/ant-maven-gradle abgerufen
- Building a RESTful Web Service. (kein Datum). Von Spring.io: https://spring.io/guides/gs/rest-service/ abgerufen
- Fejér, A. (07. 04 2019). Using Spring ResponseEntity to Manipulate the HTTP Response. Von Baeldung: https://www.baeldung.com/spring-response-entity abgerufen
- Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. Von ics.uci.edu: https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm abgerufen
- Gupta, L. (kein Datum). Spring RestTemplate – Spring REST Client Example. Von Howtodoinjava: https://howtodoinjava.com/spring-restful/spring-restful-client-resttemplate-example/ abgerufen
- Hansen, B. (09. 12 2016). Pluralsight. Von Spring Fundamentals: https://app.pluralsight.com/library/courses/spring-fundamentals/table-of-contents abgerufen
- HTTP Status Codes. (kein Datum). Von restapitutorial: https://www.restapitutorial.com/httpstatuscodes.html abgerufen

Quellen

- Kersken, S. (28. 08 2011). Openbook Rheinwerk-verlag. Von IT-Handbuch für Fachinformatiker: http://openbook.rheinwerk-verlag.de/it_handbuch/kap_10_konz_prog_005.html#8f42eaaf-0fb8-4830-9b1b-bc0fae3dd304 abgerufen
- Krischan, S. (25. 01 2019). Von developer.mozilla: https://developer.mozilla.org/de/docs/Web/HTTP abgerufen
- Pratt, M. (08. 11 2018). Get and Post Lists of Objects with RestTemplate. Von https://www.baeldung.com/spring-rest-template-list: https://www.baeldung.com/spring-rest-template-list abgerufen
- Spring Framework Overview. (31. 03 2019). Von docs.spring.io: https://docs.spring.io/spring/docs/5.1.6.RELEASE/spring-framework-reference/overview.html#overview abgerufen
- Tilkov, S. (12. 3 2009). REST – Der bessere Web Service? Von http://jaxenter.de: http://jaxenter.de/rest-der-bessere-web-service-8988 abgerufen
- Van rijn, P. (10. 10 2018). Spring REST: Getting Started. Von Pluralsighgt: https://app.pluralsight.com/library/courses/spring-rest/table-of-contents abgerufen
- Webb, P., Syer, D., & Long, J. (kein Datum). Spring Boot Reference Guide. Von docs.spring.io: https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/ abgerufen
- Winkler, R. (13.11.2018). RESTful APIs dokumentieren – so geht’s!. Von jaxenter.de: https://jaxenter.de/restful-apis-dokumentieren-52052