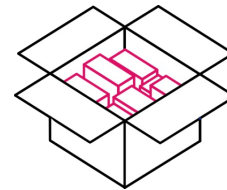


REST-ful API-Design

Hoai Viet Nguyen – TH Köln

Technology
Arts Sciences
TH Köln



Recap letzte Videos

- REST ist keine Technologie sondern ein Konzept
- Ressourcen in REST sind eindeutig identifizierbar
- Ressourcen können mehrere Repräsentationen haben
- REST-basierte System kommunizieren über ein Uniform Interface
- Lose Kopplung durch Abhängigkeit über Interface

Application Programming Interface (API)

- Schnittstelle, um Dienste (Services) nach außen anzubieten
- API ist für den Dienstkonsumenten eine Blackbox
- API ist implementierungsunabhängig
- API-Dokumentation beschreibt wie ein Client den Dienst aufrufen kann

SRP und Lose Kopplung in DDD/Spring



Web Service

- Stellt eine API als Dienst über Web-basierte Protokolle zur Verfügung
- Im Gegensatz zu Webanwendungen liefern Web Services keine UI bzw. HTML aus, sondern maschinenlesbare Daten z.B. JSON oder XML
- Typische Clients von Web Services sind z.B.:
 - Programme/Prozesse
 - Mobile App
 - IoT Geräte
 - ...

SRP und Lose Kopplung mit Web Services und DDD



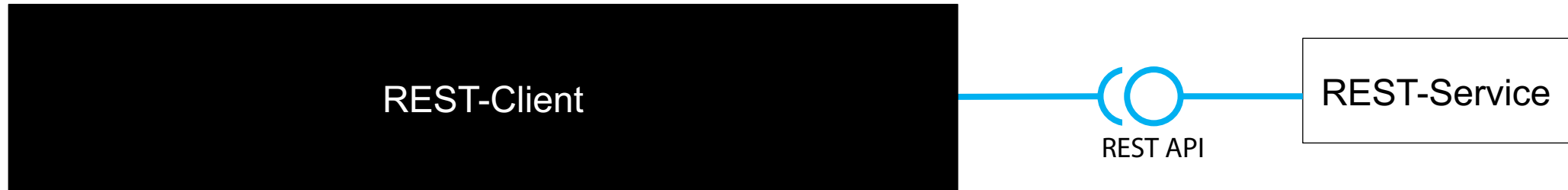
REST-APIs

- Schnittstelle für Verwendung von REST-basierten Web Services
- De-facto Standard, um Dienste im Web nach außen anzubieten
- Verschiedene Ressourcerepräsentation i.d.R. JSON
- Für REST-APIs existieren Design Best Practices

SRP und Lose Kopplung mit REST und DDD



SRP und Lose Kopplung mit REST



REST-API Design Best Practices

- **Zwei URI-Templates pro Ressource:**
 - `/resources`
 - `/resources/{id}`
- **Subressourcen:**
 - `/resources/{id}/subresources`
 - `/resources/{id}/subresources/{id}`

Konventionen zum API-Entwurf

- Verben in der URL vermeiden
- Ressource als Nomen in Plural
 - POST /users
 - GET /users
 - GET /users/{id}
 - PUT /users/{id}
 - DELETE /users/{id}

Beispiel REST-API Dokumentation

Ressource	HTTP-Verb	Content-Type (Response)	Content-Type (Request)	Statuscode
/users	POST	JSON	JSON	201 400
/users	GET	JSON		200
/users/{id}	GET	JSON		200 404
/users/{id}	PUT		JSON	204 404
/users/{id}	DELETE			204 404
/users	DELETE			204 400
...	...	...	...	...

Behandlung von Assoziationen

- **Assoziationen: Beziehungen zwischen Klassen, Beans, Entities, Daten**
- **Wie sieht z.B. der API-Aufruf aus, um alle Tasks eines Users aufzulisten?**
 - **GET /users/{id}/tasks**
- **Assoziationen (Subressourcen) werden über Konkatination realisiert**

Beispiel REST-API mit Assoziationen als Subresource

Ressource	HTTP-Verb	Content-Type (Response)	Content-Type (Request)	Statuscode
/users	POST	JSON	JSON	201 400
/users	GET	JSON		200
/users/{id}	GET	JSON		200 404
/users/{id}	PUT		JSON	204
/users/{id}	DELETE			204
/users	DELETE			204 400
/users/{id}/tasks	POST	JSON	JSON	201 404 400
/users/{id}/tasks	GET	JSON		200
/users/{id}/tasks/{taskId}	GET	JSON		200 404
...	...	...	...	...

Wie werden komplexe Assoziationen bzw. Filter behandelt?



Beispiel REST-API mit komplexen Assoziationen bzw. Filter

Ressource	HTTP-Verb	Content-Type (Response)	Content-Type (Request)	Statuscode
/users	POST	JSON	JSON	201 400
/users	GET	JSON		200
/users/{id}	GET	JSON		200 404
/users/{id}	PUT		JSON	204
/users/{id}	DELETE			204 404
/users	DELETE			204
/users/{id}/tasks	POST	JSON	JSON	201 400
/users/{id}/tasks	GET	JSON		200 404
/users/{id}/tasks?open=true false	GET	JSON		200 404
/users/{id}/tasks/{taskId}	GET	JSON		200 404
...	...	...	...	...

Datenformate für Ressourcerepräsentationen

- Wie wählt man den Abruf eines bestimmten Datenformat einer Ressource aus?
 - Accept: <MIME-Type>
 - z.B Accept: application/json
- Im Webbrowser ist es nicht möglich Header selber zu setzen. Wie wählt man dort die Ressourcerepräsentation aus?
 - /users/john/comments?mime-type=json
 - /users/john/comments.json

Suche

- Globale Suche
 - `/events?q=rock`
- Fokussierte Suche
 - `/events/rar2016/acts?q=johnossi`

Pagination und Begrenzung

- Offset und Limit
 - `/events?limit=10`
 - `/events?limit=25&offset=50`
- Page
 - `/tweets?page=3`
- Start und Ende
 - `/videos/{id}?start=123s`
 - `/videos/{id}?start=10s&end=50s`

Sortierung

- `/events?sort=desc`
- `/events?sort=asc`

Versionierung

- API-URL

- `/v1/users` Mein Favorit

- `/users?version=1`

- Header

- `Accept-Version: 1`

- `Accept: application/vnd.myname.v1+json`

Diese Fragen sollten Sie nach der heutigen Sitzung beantworten können

- **Wie entwirft man eine REST-API?**
- **Wie geht man mit Assoziationen in REST-APIs um?**

Zusammenfassung REST-API Design

- Nur zwei URLs pro Ressource
- Keine Verben in der URL
- CRUD über HTTP-Methoden
- Nomen in Plural
- Komplexität hinter das Fragezeichen “?”
- Assoziationen durch Konkatenation der Subressource

Danksagung

Die verwendeten Icons stehen unter der Flaticon Basic Lizenz (flaticon.com) und stammen von

- Freepik
- Atif
- juicy fish
- BZZRINCANTATION