

1 Anforderungen

1.1 Funktionale Anforderungen

Nr.	Anforderung	Priorität
1	Das System muss den Zugriff auf die neuen, verlinkten Datenstrukturen von swissbib mit einer REST Schnittstelle bereitstellen.	1
1.1	Die REST Schnittstelle soll unter http://data.swissbib.ch öffentlich angeboten werden	
1.2	Die REST Schnittstelle soll die Grundsätze (Stateless, Cacheable, Client-Server, Layered System und Uniform Interface) befolgen.	
2	Das System muss eine einheitliche Schnittstelle nach aussen bieten.	1
2.1	Eine Definition der erlaubten URLs wird erarbeitet und dokumentiert.	
2.2	Eine Definition der erlaubten Parameter für eine Filterung der Suche über die URL wird erarbeitet und dokumentiert.	
2.3	Eine Definition der erlaubten HTTP-Header wird erarbeitet und dokumentiert.	
3	Das System muss eine Abstraktion gegen innen bieten, die es ermöglicht verschiedene Backend Services (z.B. Datenbanken, Suchmaschinen) einfach zu integrieren.	1
3.1	Ein Interface für Backend Services muss definiert werden.	
3.2	Weitere Backend Service sollen per Konfiguration hinzugefügt werden können. (Ohne Änderungen am sonstigen Code)	
4	Die Backend Services sollen für Clients des Systems transparent sein.	2
4.1	Backend Services dürfen die öffentliche Schnittstelle nicht verändern.	
5	Das System muss den bestehenden Elasticsearch Server als Backend integrieren.	1
5.1	Es soll eine unabhängige Komponente für die Abfragen auf den bestehenden Elasticsearch Server implementiert werden.	
5.2	Die Komponente soll für das Projekt relevante Teile der Query DSL implementieren.	
5.3	Suchabfragen können mittels Konfigurationsdateien beeinflusst werden.	
6	Das System soll Content Negotiation für folgende Formate unterstützen: HTML, JSON-LD, Turtle, RDF/XML und NT.	2
6.1	Das Format soll über den Accept Parameter im Header vom GET-Request bestimmt werden können	
6.2	Das Format soll über die URL bestimmt werden können	
6.3	Die HTML Repräsentierung soll auch für Clients als verlinkte Daten interpretierbar sein.	
7	Das System muss sich in die Serviceplattform von swissbib auch im produktiven Betrieb integrieren lassen.	1
7.1	Die Implementierung muss mit der Programmiersprache PHP (Version 7) erfolgen.	
7.2	Die Applikation soll mit einem Apache Webserver betrieben werden können.	
8	Das System soll Endpunkte für folgende Ressourcen anbieten: bibliographicResource, documents, organisation, work, item, person	2
8.1	Von jeder Entität soll eine Collection abgefragt werden können.	

8.2	Von jeder Entität sollen Einträge mit einer bestimmten ID abgefragt werden können.	
8.3	Eine Suche mit Filterfunktion für einzelne Entitäten soll angeboten werden	
8.3.1	Eine Suche mit Filterfunktion für einzelne Entitäten beschränkt auf definierte Felder soll angeboten werden	
8.4	Eine Suche mit Filterfunktion über mehrere / alle Entitäten kann angeboten werden	
8.5	Jedes Attribut wird eindeutig einem Vokabular zugewiesen.	
8.6	Jedes Attribut kann mit @type (z.B. @id, Literal) einem bestimmten Datentyp zugewiesen werden.	
9	Das System soll komponentenbasiert aufgebaut werden.	2
9.1	Abhängigkeiten zwischen Komponenten sollen über ein Dependency Managementsystem verwaltet werden.	
9.2	Die Komponente, die unter Punkt 5 definiert ist soll ebenfalls über das Dependency Management integriert werden.	
10	Das System muss dokumentiert sein.	1
10.1	Der Code ist ausführlich mit Kommentaren versehen.	
10.1.1	Code Kommentare folgen den PHPdoc Vorgaben.	
10.2	Das System soll einen Hydra doc Endpunkt zur Verfügung stellen.	
10.3	Das System bietet eine HTML Dokumentation für die Nutzer der Schnittstelle.	
10.4	Die Softwarearchitektur soll dokumentiert werden.	
11	Das System muss nach den Prinzipien von Hydra umgesetzt werden.	1
11.1	Das System soll die Hydra Konzepte Endpoint, Doc, Collection und Error unterstützen.	
11.2	Das System bietet eine Paginierung für Collections	

1.2 Nicht funktionale Anforderungen

Nr.	Anforderung	Priorität
90	Für das System sollen Performance Analysen gemacht werden.	2
90.1	Die Performance des Gesamtsystems soll analysiert werden.	
90.2	Die Performance der einzelnen Komponenten der Applikation soll analysiert werden.	
91	Das System muss definierte CodeStyle Regeln einhalten.	1
92	Wichtige Teile des Systems sollen mit Unit Tests automatisiert getestet werden.	2
92.1	Die Tests sollen automatisiert auf einem Continuous Integration Server ausgeführt werden.	
93	Das System soll mit zukunftsorientierten Technologien erstellt werden.	2
94	Das System kann mit Lasttests getestet werden.	3

Das System = Rest API

Prioritäten	1 = Muss	2 = Soll	3 = Kann
-------------	----------	----------	----------

2 Szenarien

Unsere REST Schnittstelle kann als Applikation als Push-Applikation eingestuft werden. Das bedeutet, dass sie nicht auf eine Nachfrage erstellt wird, sondern dass wir mit dieser Hypermedia driven REST API eine Nachfrage generieren wollen. Dies bedeutet für uns, dass spezifische Anwendungsszenarien nur geschätzt werden können, denn es existiert momentan noch kein akutes Bedürfnis nach einer solchen Schnittstelle bei swissbib.

Die REST Schnittstelle muss folgende Hauptszenarien sicherstellen.

1. Sie ist 'der' Zugriffspunkt für maschinelle Clients, welche die innerhalb des Projekts linked.swissbib.ch modellierten und erstellten Datenstrukturen nutzen wollen.
2. Basis für interne/externe Webapplikationen, welche semantische Daten in ihrer Domäne nutzen wollen. Die Rest Schnittstelle bietet weltweit eindeutige Ids, mit der die Ressourcen abgefragt werden können.
3. Die REST Schnittstelle soll es uns ermöglichen, bestimmte Regeln für den Zugriff auf die Daten zu definieren (Filterfunktion).
4. Der entwickelte Adapter soll auch von anderen Applikationen wiederverwendet werden können. Beispielsweise kann das OpenSource Projekt VuFind (<http://vufind.org/demo/>) bisher zwar einen SOLR Suchserver nutzen aber noch keinen Elasticsearchserver. Mithilfe des neuen Adapter soll dies auch möglich sein.