

A Concept for the Design of Learning Resources for APIs of Content Management Platforms

Guided research – Final Presentation

Sirma Gjorgjevska, 23.05.2016

Software Engineering for Business Information Systems (sebis)

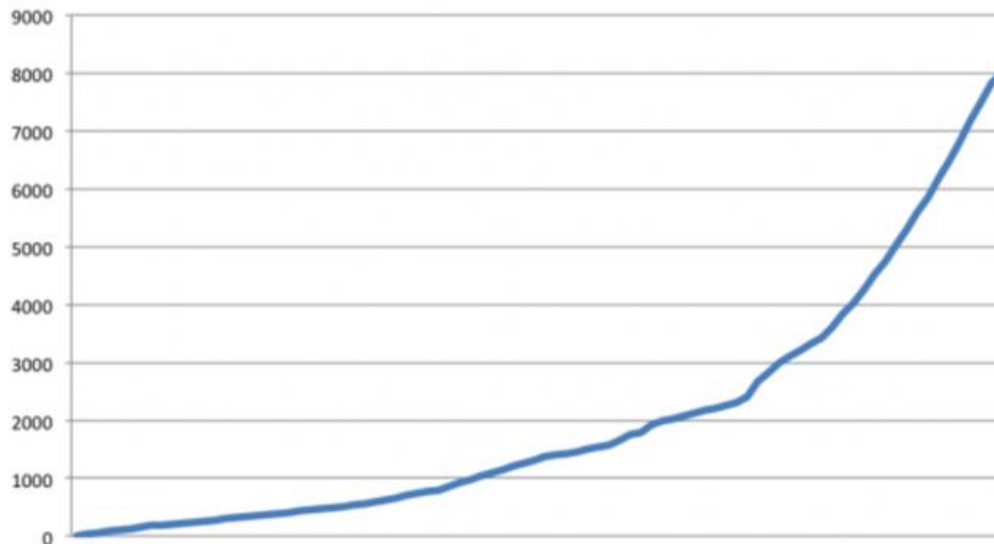
Department of Informatics

Technische Universität München, Germany

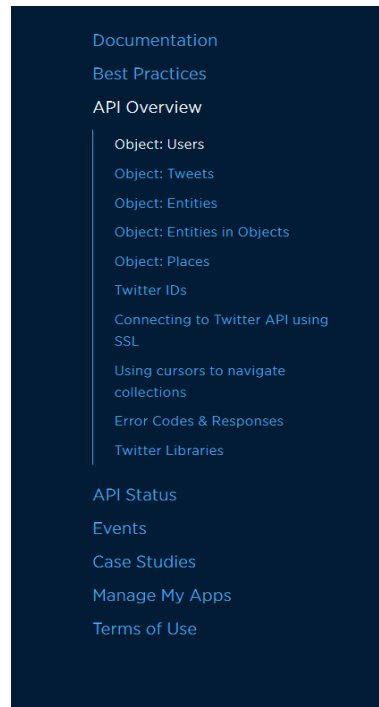
www.matthes.in.tum.de

- 1. Introduction**
- 2. Problem statement**
- 3. Shortcomings of current API learning resources**
- 4. A conceptual model for API documentation**
- 5. Prototype**

- **Application-programming interfaces (APIs)**
 - Allow one program or web site to access the data and services provided by another program or website
 - Make programming easier
- The number of web applications offering APIs has increased dramatically



- **API documentation can be critical for software developers**
- Good documentation can help developers work efficiently
- Documentation that doesn't meet the readers' expectations can lead to
 - Frustration
 - Major loss of time



Users

Users can be anyone or anything. They [tweet](#), [follow](#), [create lists](#), have a [home_timeline](#), can be [mentioned](#), and can be [looked up](#) in bulk.



Field Guide

Consumers of Users should tolerate the addition of new fields and variance in ordering of fields with ease. Not all fields appear in all contexts. It is generally safe to consider a nulled field, an empty set, and the absence of a field as the same thing.

Field	Type	Description
<code>contributors_enabled</code>	Boolean	Indicates that the user has an account with "contributor mode" enabled, allowing for Tweets issued by the user to be co-authored by another account. Rarely <code>true</code> . Example:

```
"contributors_enabled": false
```

“Most documentation is written by technicians-not professional writers. And most technicians would include documentation among their top ten complaints regarding the software they use.”

“Some of the most severe obstacles faced by developers learning new APIs pertained to the documentation and other learning resources.”

“Documentation is often poorly written and finding useful content in documentation can be so challenging that people might not try to do so.”

“Users do not read documentation. Instead, they usually first try other possible methods of finding solutions. This suggests that current approaches to developing and delivering documentation may not be providing the solution paths that users seek”

“The problem is not that people cannot follow simple steps; it is that they do not. People are thrown into action; they can only understand through the effectiveness of their actions in the world. People are always already trying things out, thinking things through, trying to relate what they already know to what is going on, recovering from errors.”

1. Introduction
2. Problem statement
3. Shortcomings of current API learning resources
4. A conceptual model for API documentation
5. Prototype



What are the current approaches for learning an API?
What are their strengths and weaknesses?



What are the principles for designing learning resources?

1. Introduction
2. Problem statement
3. Shortcomings of current API learning resources
4. A conceptual model for API documentation
5. Prototype

- Out of date documentation
- Incomplete and incorrect documentation
- Unclear description of API elements - ambiguous documentation
- Documentation that fails to provide good navigation
 - Developers have difficulty locating solutions to problems quickly
- Poor presentation of API elements
 - Description is verbose or excessively extensive
 - Information for a specific element is fragmented over many pages

1. Introduction
2. Problem statement
3. Shortcomings of current API learning resources
4. A conceptual model for API documentation
5. Prototype

1

Up to date documentation

- Documentation should reflect the most recent changes

2

Documentation of the API's high level design

- Explains the architecture that is used for developing the API

3

Quickstart

- Instructions for installing and setting up the development environment

4

Tutorial

- Interactively try out the API invocations

5

Best practices

- Examples that show the best practices of an API's use

6

API reference

- List of all API's functionalities

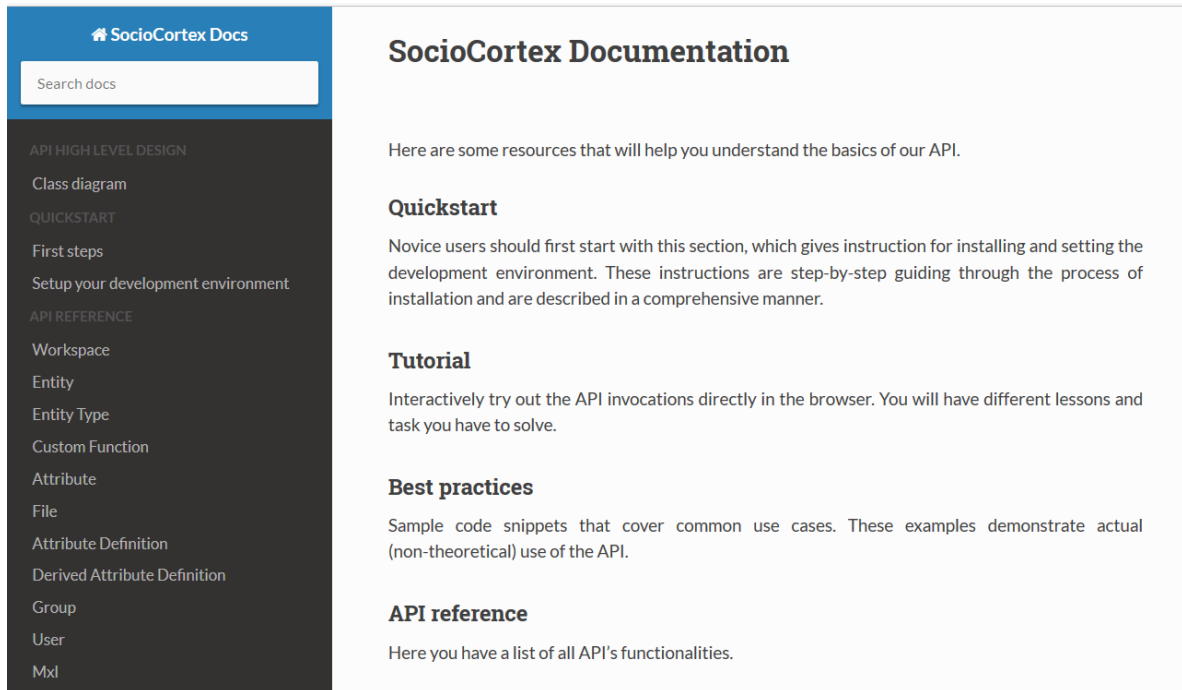
7

Multiple ways of navigation

- Overview page, categories, full text search and link to related resources

1. **Introduction**
2. **Problem statement**
3. **Shortcomings of current API learning resources**
4. **A conceptual model for API documentation**
5. **Prototype**

- Documentation for Content Management Platform SocioCortex
- AngularJS web application that integrates with SocioCortex backend
- Read The Docs framework for the look and the feel of the web application
- Demo



The screenshot shows the SocioCortex Documentation website. The left sidebar contains a search bar and a navigation menu with the following items: API HIGH LEVEL DESIGN, Class diagram, QUICKSTART, First steps, Setup your development environment, API REFERENCE, Workspace, Entity, Entity Type, Custom Function, Attribute, File, Attribute Definition, Derived Attribute Definition, Group, User, and Mxl. The main content area is titled "SocioCortex Documentation" and contains the following sections:

- SocioCortex Documentation**

Here are some resources that will help you understand the basics of our API.
- Quickstart**

Novice users should first start with this section, which gives instruction for installing and setting the development environment. These instructions are step-by-step guiding through the process of installation and are described in a comprehensive manner.
- Tutorial**

Interactively try out the API invocations directly in the browser. You will have different lessons and task you have to solve.
- Best practices**

Sample code snippets that cover common use cases. These examples demonstrate actual (non-theoretical) use of the API.
- API reference**

Here you have a list of all API's functionalities.

Thank you for your attention!



Sirma Gjorgievska

B.Sc

sirma.gjorgievska@tum.de



Technische Universität München
Department of Informatics
Chair of Software Engineering for
Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289.
Fax +49.89.289.17136

www.matthes.in.tum.de

- [1] Rettig, M. 1991. **Nobody reads documentation.** *Commun. ACM*, 34(7): 19-24.
- [2] David G. Novic, K. W. 2006. **What users say they want in documentation,** *SIGDOC'06*.
- [3] Watson, R., Stamnes, M., Jeannot-Schroeder, J., & Spyridakis, J. H. 2013. **API documentation and software community values: a survey of open-source API documentation.** Paper presented at the Proceedings of the 31st ACM international conference on Design of communication.
- [4] Buse, R. P., & Weimer, W. 2012. **Synthesizing API usage examples.** Paper presented at the Software Engineering (ICSE), 2012 34th International Conference.
- [5] Robillard, M. P. 2009. **What Makes APIs Hard to Learn. Answers from developers.** *IEEE Computer Society*.

- [6] Earle, R. H., Rosso, M. A., & Alexander, K. E. 2015. **User preferences of software documentation genres**, *Proceedings of the 33rd Annual International Conference on the Design of Communication*: 1-10. Limerick, Ireland: ACM.

- [7] Robinson, P. J. 2011. **MyPyTutor: an interactive tutorial system for Python**, *Proceedings of the Thirteenth Australasian Computing Education Conference - Volume 114*: 155-160. Perth, Australia: Australian Computer Society, Inc.

- [8] Uddin, G., & Robillard, M. P. 2015. **How API Documentation Fails**. *Software, IEEE*, 32(4): 68-75.

- [9] Danielsen, P. J., & Jeffrey, A. 2013. **Validation and interactivity of Web API documentation**. Paper presented at the Web Services (ICWS), 2013 IEEE 20th International Conference.