

Evaluation – Collaboration-Frameworks Projekt: DigitalSchoolNotes

Projekt Team: Adler, Brinnich, Hohenwarter, Karic, Stedronsky

Version 1.1

12.10.2015

Status: [RELEASE]

	Datum	Name	Unterschrift
Erstellt	07.10.2015	ThomasStedronsky	
Geprüft	07.10.2015	Adin Karic	
Freigegeben			
Git-Pfad: /doc/evaluation Dokument: evaluation_collaboration.doc			

1 Changelog

Version	Datum	Status	Bearbeiter	Kommentar
0.1	2015-10-07	Erstellt	Selina Brinnich	Dokument erstellt
1.0	2015-10-07	Geprüft	Niklas Hohenwarter	QA
1.1	2015-10-12	Bearbeitet	Thomas Stedronsky	Google Drive, Formulierungen überarbeitet, Abbildungsverzeichnis, Inhaltliche Korrekturen bei 1. und 2. Code Screenshots in Text umgewandelt.
2.0	2015-10-12	Geprüft	Adin Karic	QA

2 Vergleich

	ape Project	TogetherJS
Dokumentation	9/10 Sehr ausführlich, genaue Einsicht in die Struktur	7/10 Ist vorhanden, allerdings keine richtige API sondern lediglich eine Auflistung wie einzelne Funktionen funktionieren
Bedienung	10/10 In der Demo gut ersichtlich, wie der APE Server arbeitet und leicht nachzumachen	6/10 Vom Prinzip her relativ leicht aber es ist nicht für unsere Anwendungen gemacht. Es kann nur die ganze Seite teilen und nicht einzelne Elemente
Lizenz	10/10 Darf kostenlos kommerziell genutzt werden. Mit einer GNU License [1]	10/10 Darf kostenlos kommerziell genutzt werden.
Installation	8/10 Installation laut Tutorial [2] einfach nachzumachen.	10/10 Muss nicht installiert werden läuft über den Mozilla Server. Allerdings besteht auch die Möglichkeit diesen Server selber zu hosten
Community	5/10 Fast nicht vorhanden. 683 Fragen bei Stackoverflow	3/10 Kaum vorhanden 132 Fragen bei Stackoverflow
Eigene Elemente definieren	10/10 Können integriert werden. Mittels einfachen <div> Containern	0/10 Nicht ersichtlich
Prototyp	10/10 Demos wurden zum Laufen gebracht. Das Chat Example funktioniert zum Beispiel mit 2 Browsern.	3/10 Nur der online Prototyp von der Herstellerwebsite
Summe	61/70	39/70

3 Ape Project

APE Server und das JavaScript Framework wurden von [3] und [4] heruntergeladen und anschließend auf einen Linux Debian 8 installiert. [1]

Mittels `dpkg -i APE_Server-1.1.2-amd64.deb apt-get -f install` kann das Paket installiert werden. Das APE JavaScript Framework kann mittels Download und anschließendem `unzip` installiert werden.

Den kompletten Ordner konnte man hierbei in einen beliebigen Ordner verschieben, man muss lediglich den Pfad im Virtuellen Host angeben (genauere Information folgt).

Hier die Erklärung wie man einen Virtuellen Host in nginx erstellen kann.

In `/etc/nginx/sites-available`

```
server {
    listen 8080;
    server_name *.ape.local;

    root /etc/ape/APE_JSF;
    index index.html

    location/ {
        try_files $uri $uri/ =404;
    }
}
```

Danach wurde in `/etc/ape/ape.conf` der Server konfiguriert.

```
server {
    port = 80
    daemon = yes
    ip_listen = 0.0.0.0
    domain = auto
    rlimit_nofile = 10000
}
```

Außerdem wurde im nächsten Schritt die DNS Konfiguration gemacht.

Hierbei mussten in dem Host File des Hosts diverse Domains eingetragen werden.

In Windows:

```
C:\Windows\System32\drivers\etc\hosts
127.0.0.1 ape.local
192.168.13.132 ape.local
192.168.13.132 0.ape.local
192.168.13.132 1.ape.local
192.168.13.132 2.ape.local
192.168.13.132 3.ape.local
192.168.13.132 4.ape.local
192.168.13.132 5.ape.local
192.168.13.132 6.ape.local
192.168.13.132 7.ape.local
192.168.13.132 8.ape.local
192.168.13.132 9.ape.local
```

In Linux (Server) `/etc/hosts`

Danach musste die im Schritt zuvor erstellte URL in die Config Files der Demos eingefügt werden. Dies passiert in dem config File der JS Demos.

/etc/ape/APE_JSF/Demos/config.js

Hier gab es 3 Punkte die ausgefüllt werden mussten.

```
APE.Config.baseUrl = 'ape.local:8080/APE_JSF';
```

```
APE.Config.domain = 'ape.local:8080';
```

```
APE.Config.server = 'ape.ape.local:8080';
```

Nach diesem Schritt musste der Server neugestartet werden. Dies geschieht mit `service ape-server restart` bzw. `service ape-server stop` und dann `service ape-server start`

Danach sollte die Installation getestet werden, dies geht mit *ape.local/Tools/Check*

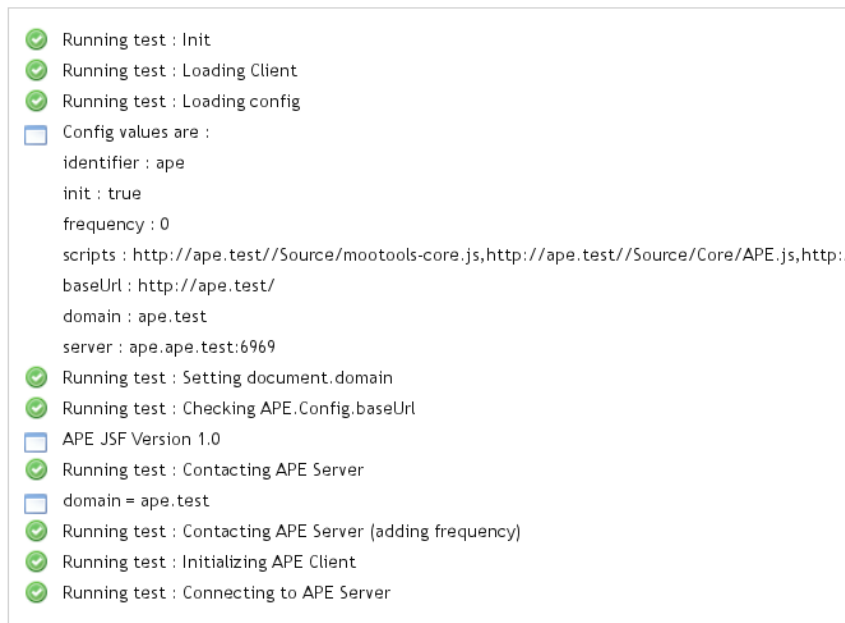


Figure 1: running test ape

Anschließend wurde mit *ape.local:8080/Demos/demo.html* die Chat Demo ausgeführt. Dies wurde gleichzeitig in 2 verschiedenen Browsern ausgeführt.

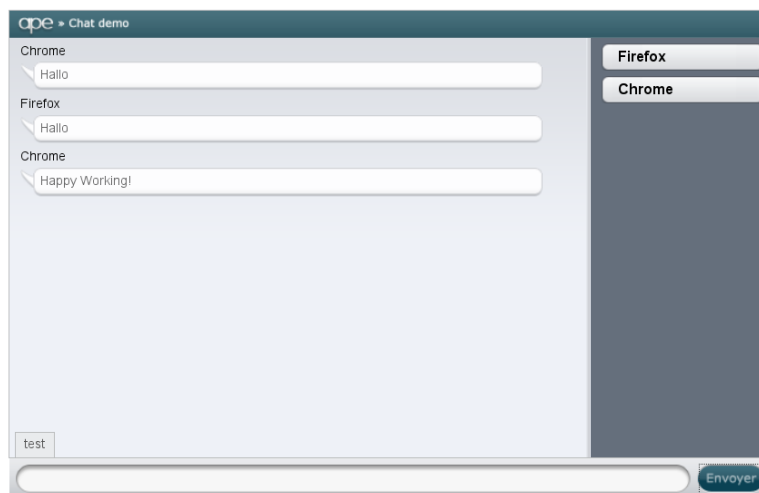


Figure 2: chat prototyp ape

4 TogetherJS

Es entspricht nicht unseren Anforderungen, weil es eine eigene Konsole ist und es bezieht sich nicht auf einzelne Elemente sondern auf die ganze Website. Man kann das Framework nicht auf einzelne Elemente beschränken. Hierbei besteht das Problem das nur eine ganze Seite geteilt werden kann, das ist für unsere Zwecke nicht zielführend, weil das Ziel von digitalschoolnotes.com ist einzelne Elemente mit anderen Usern zu teilen. [5]

5 Derby

Kann für unsere Zwecke nicht verwendet werden, da es ein Full-Stack Framework ist. Hierbei wurden, bei der anfänglichen Recherche der zu evaluierenden Frameworks Fehler gemacht. [6]

6 GoogleDrive

Google verwendet die eigens erstellte Google Realtime API. Diese API ist allerdings nicht öffentlich zugänglich und kann daher auch nicht genutzt werden. [7]

7 Endergebnis

Aufgrund der Ergebnisse der Evaluierungstabelle, sowie den zusätzlichen Erkenntnissen durch die Prototypen entscheidet das Team sich dazu **APE Project** zu verwenden.

8 Quellen

[1], APE (Ajax Push Engine), "Comet server Real time data streaming",
<http://ape-project.org/>, zuletzt besucht: 07.10.2015

[2], APE-Project, "APE_Server",
https://github.com/APE-Project/APE_Server/wiki/APE-Server-install, zuletzt besucht: 05.10.2015

[3], APE, "Download APE (Ajax Push Engine)",
http://ape-project.org/download/stable/APE_Server-1.1.2-amd64.deb, zuletzt besucht: 07.10.2015

[4], APE, "Download APE (Ajax Push Engine)",
http://ape-project.org/download/stable/APE_JSF-1.1.0.tar.gz, zuletzt besucht: 07.10.2015

[5], mozilla, "togetherjs",
<https://github.com/mozilla/togetherjs>, zuletzt besucht: 05.10.2015

[6], derby, "Derby Getting Started",
<http://derbyjs.com/started>, zuletzt besucht: 01.10.2015

[7] Google Developers, „Realtime API“,
<https://developers.google.com/googleapps/realtime/overview>, zuletzt besucht: 12.10.2015

9 Abbildungsverzeichnis

Figure 1: running test ape.....	5
Figure 2: chat prototyp ape.....	5