

# Building APIs

- × When does your project need an API?
  - × Rapidly changing data
  - × Large volume of requests
  - × Many people find it useful and want to build on top of it or extend it
  - × You want to build on top of your data
  - × Trust issues



# Planning

- × Stable data structure
- × Think about your audiences: users and developers
  - × How will they use it? (Features and requirements - what operations?)
  - × What should be exposed?
  - × How will users authenticate?
- × Infrastructure: where will the API be hosted? Same server with subdomain, different machine? Will it scale?
- × [Best Practices Article](#)
- × Testing and Development: [Postman](#) is a useful tool



Web Editor
View
Compare
Current document: manuscript/codex/vorGoetsJm1
dcd:111

---

Close

Filter objects:

[Add new object from URL: http://...](#)

| Cod. 133 F 17, fol. 162ra-168rb |  |
|---------------------------------|--|
|                                 |  |
|                                 |  |
|                                 |  |
|                                 |  |
|                                 |  |
|                                 |  |
|                                 |  |

12 items

| Cod. 814, fol. 230ra-233rb |  |
|----------------------------|--|
|                            |  |
|                            |  |
|                            |  |
|                            |  |

8 items

| Ms Germ. Quart. 524, fol. 159r-163r |  |
|-------------------------------------|--|
|                                     |  |
|                                     |  |
|                                     |  |
|                                     |  |
|                                     |  |

9 items

| Ms Germ. Quart. 261, fol. 186r-190v |  |
|-------------------------------------|--|
|                                     |  |
|                                     |  |
|                                     |  |
|                                     |  |
|                                     |  |

10 items

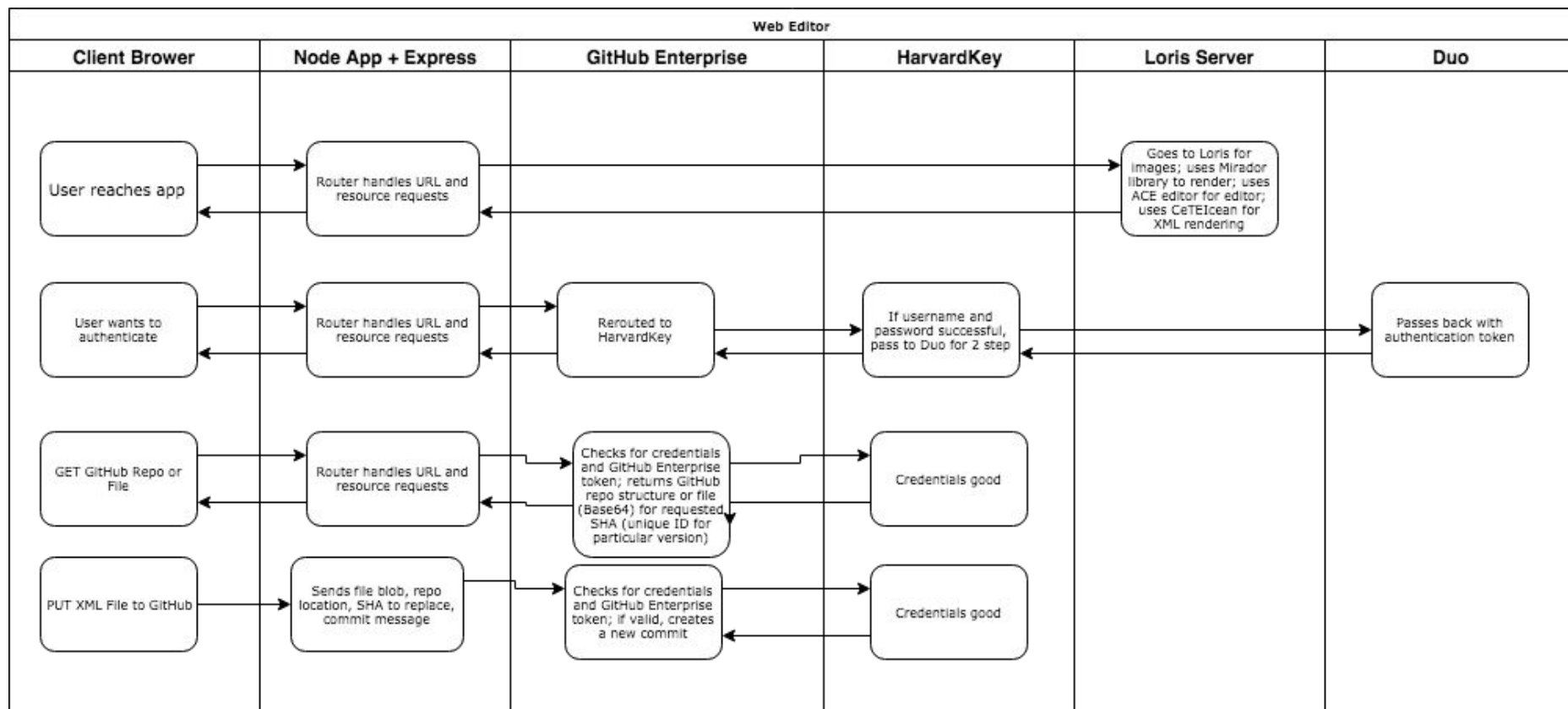
Below is a rendering of the page up to the first error.

```

123 -----><|b>ue schone lichen</b>um sol werden en.
124 -----><|b>siste der nome
125 -----><|b>de antwende auch edref="R0" facs="R6_163r-a">x</cb><b>ria ende sprac mjn.
126 -----><|b>lle euerenre.
127 -----><|b>suster, na dien dot.
128 -----><|b>ic som kin nader.
129 -----><|b>werlt Die sude ic.
130 -----><|b>v weest te vreden ende.
131 -----><|b>let my min goet.
132 -----><|b>stren mit gheleche<b>n</b>ochten, ende witten.
133 -----><|b>moens deser werlt.
134 -----><|b>de leet om vrolic.
135 -----><|b>wissen, wie weet hoe.
136 -----><|b>nghe doet<b>en</b>em sal, des enden.
137 -----><|b>dacht noch name type="person" myRef="Mary_Magdalena" rend="nub_underline">maris</name>.
138 -----><|b>thus inde name type="person" myRef="Martha" rend="nub_underline">martha</name>.
139 -----><|b>reyne dat hoor suster.
140 -----><|b>so wilt wi, doer<b>in</b> ouer lee seer<b>in</b> hadden die linc<b>in</b>kers ende constenen.
141 -----><|b>vee name type="place" myRef="Ierusalem">ierusalem</name> een.
142 -----><|b>ker beropen ende.
143 -----><|b>willie daer wert.
144 -----><|b>maria neglene.
145 -----><|b>ke speli ghenen inde.
146 -----><|b>al quon daer mit.
147 -----><|b>spel clertelt ende.
148 -----><|b>eb edref="R0" facs="R6_163r-v">ygh<b> ed</b> edref="R0" facs="R6_163r-a">x</cb>.
149 -----><|b>ssal<b>el</b> ual rend="strickthrough">nader werlt<b>els</b>.
150 -----><|b>kers liches als.
151 -----><|b>nen nynter deer.
152 -----><|b>werlt inde si dande.
153 -----><|b>ende sprac so monde.
154 -----><|b>ree ghien linc, so.
155 -----><|b>dat si naue ipse.
156 -----><|b>men en machte.
157 -----><|b>In dese saluen tiden.
158 -----><|b>sal vernae name type="person" myRef="Martha" rend="nub_underline">martha</name>.
159 -----><|b>her suster dat onsen.
160 -----><|b>her name type="person" myRef="JesusChrist" rend="nub_underline">ihus</name>wes inden.
161 -----><|b>inde van name type="place" myRef="Gallilee">gillilee</name>.
162 -----><|b>nde end tot hoer.
                    
```

(column break)

Hier beghynt die his  
tone van die bekenn  
ghe der heiligher vrien  
dynnen ende apostelyn  
na 9009 ante maria mag  
dalene Ade als ydco  
rus bescreuen heeft  
Maria  
was  
gebor  
en van  
edelen  
gheslachten ende stamme  
ende eerlike gebor  
ten der stat van ihe  
rualem haer vader  
hie cyrus haer  
moeder hietu nucha  
rie ende si hadde twee  
dochteren ende einen zoen  
Die een dochter hiet  
martha, die ander  
marc. Die zoen hei  
te lasarus dese was



# Example: Harvard Art Museums

- × Documentation:  
<https://github.com/harvardartmuseums/api-doc>
- × Register: <https://www.harvardartmuseums.org/collections/api>
- × API App Examples
  - × <http://apps.harvardartmuseums.org/art-explorer/>
  - × <http://apps.harvardartmuseums.org/museum-explorer/>
  - × <http://apps.harvardartmuseums.org/suns-explorer/>



# Example: Blog Application

NodeJS + Express

<http://159.203.177.188:3000/posts>

<http://159.203.177.188:3000/posts/api/2>

Python + Flask Resources

<http://programminghistorian.github.io/ph-submissions/lessons/creating-apis-with-python-and-flask/>

<https://auth0.com/blog/developing-restful-apis-with-python-and-flask/>

```
var express = require('express');
var router = express.Router();

router.post('/submitpost', (req, res, next)=>{
  req.app.locals.postcount++;
  var postcount = req.app.locals.postcount;
  var postdata = {
    "title": req.query.title,
    "author": req.query.author,
    "date": req.query.date,
    "content": req.query.content,
    "image": req.query.picture,
    "postid": postcount
  };
  req.app.locals.posts[postcount] = postdata;
  res.render('post', postdata);
});

router.get('/:postid', (req, res, next)=>{
  res.render('post', req.app.locals.posts[req.params.postid]);
});

router.get('/api/:postid', (req, res, next)=>{
  res.write(JSON.stringify(req.app.locals.posts[req.params.postid]));
  console.log('/api/postid');
  res.end();
});

router.get('/', (req, res, next)=>{
  res.render('posts', {
    "posts": req.app.locals.posts
  });
});

module.exports = router;
```



# CMS REST APIs

- × Easiest way to add an API to your website
  - × CMSes already force some structure on your content
- × Omeka
  - × <http://omeka.readthedocs.io/en/latest/Reference/api/>
- × Scalar
  - × <http://scalar.usc.edu/works/guide2/working-with-the-api>
- × WordPress
  - × Now in core: <https://developer.wordpress.org/rest-api/>
  - × Example: <http://darthcrimson.org/wp-json/wp/v2/posts>
- × Drupal
  - × 7: <https://www.drupal.org/project/restws>
  - × 8: REST is in core

