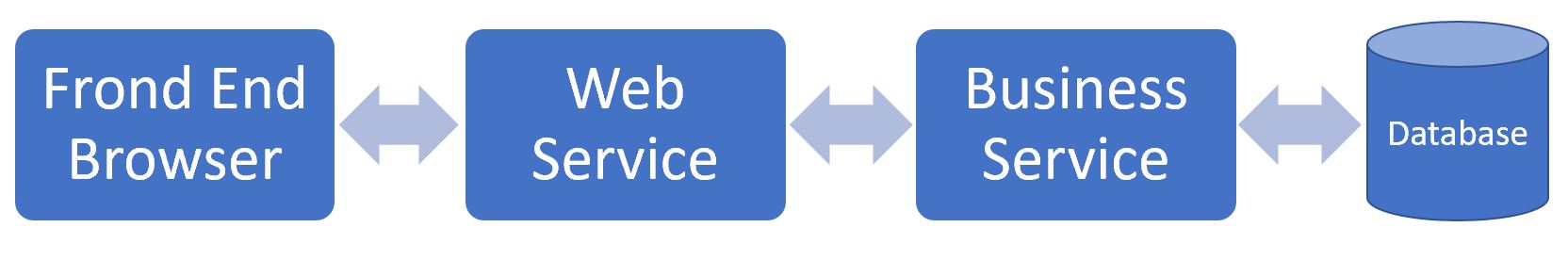
# How to build a website

With the help of public cloud, it’s much easier to setup a website. This article will show you how to build a website.



There are four main layers for a website, frond end, web service, business service and database. Frond end is for visualization and show user interface to customer. Frond end is the essential part for most of website, excepting providing API services. For the simple static website, front end could be the only part which is required. The other three are from back end. Web service is to accept frond end request and return related information/web page to frond end. The business service is to hold the business logic which could be one business service component or multiple business services which are always the case for complex web site. The last but not the least is done storage database. The database is to persist the webservice status, business data, user’s input etc.

## Frond end

The main part for the frond end technology is html, css and java script. Duke university provide a great open course about html, css and java script. <https://www.coursera.org/learn/duke-programming-web>. W3school also provides tutorials about html, css, js etc. <https://www.w3schools.com/>. Codepen is a great tool to test webpage <https://codepen.io/anon/pen/pxmZVM?&editors=0010>. Dust (<http://www.dustjs.com/>) is one technology to provide html template. For this tutorial, we only use basic html, css and js with jQuery.

## Web Service

There are many different technologies to build web service, such as php, jsp, asp, nodejs etc. Nodejs ([https://nodejs.org](https://nodejs.org/) ) is selected as it’s the newest technology for building web service. Kraken ([http://krakenjs.com](http://krakenjs.com/)) is one framework that is based on nodejs. In this tutotial, express (<https://expressjs.com/>) and react (<https://reactjs.org/docs/introducing-jsx.html>) are used. Other resources

<https://stackoverflow.com/questions/47371904/e-unable-to-locate-package-npm> and

<https://code.visualstudio.com/docs/nodejs/nodejs-tutorial>.

## Business service

There are many ways to build business service. In this tutorial, we use java based spring boot technology to build restful service with json format.

## Development environment.

As webstorm is not free, visual studio code is best free web integrated development environment. Visual studio code is also for nodejs web service.

For the back-end java service, intelliJ is recommended IDE as eclipse is too slow and heavy.

## Deployment environment.

After the code is done and tested locally, it’s ready to be deployed so the rest of the world could access it. There are several places the service could be deployed. The traditional one is to deploy to rent or buy one or several machine in IDC and deploy the components to it. You can also buy machines and put it in your home, and deploy the services to the machines. Any way you need apply a public IP, so other people could access it from any place with internet connection. Public cloud provides convenience and one stop solution for hosting services. Amazon aws and google cloud are top two public clouds that being used by difference purpose to host web/business services. The existed app engine can be used to deploy component or the components can be wrapped to docker and then deployed dockized component to Amazon aws or google cloud.

Some documents about deployment.

Deploy: <https://www.youtube.com/watch?v=n4svrNcAkJg>

<https://stackoverflow.com/questions/15444020/cannot-find-module-findup-sync-when-trying-to-run-grunt>

<https://cloud.google.com/storage/docs/hosting-static-website>

<https://github.com/jooyeong/kraken>

Docker: <https://medium.com/@cramirez92/build-a-nodejs-cinema-microservice-and-deploying-it-with-docker-part-1-7e28e25bfa8b>

## Domain name

In order for the rest of the world access your webservice easier, domain name is required which is easy to remember rather than an ip address. I recommend to use google domain if you use google cloud, or Amazon router 35 if aws is used for web service. Otherwise godaddy is traditional web site to apply domain name. Use the same domain provider as the web service, will save you tons of time to config it.

## Analytics

Google analytics (<https://analytics.google.com/>) is easy and free way to integrate for web site customer access analysis.

## Marketing

Google search engine and bing search engine need to be added for SEO (Search engine optimization). Both search engines need site map, <https://www.xml-sitemaps.com/> is a free and easy to use online site map generate tool.

Good adsense (<https://www.google.com/adsense/start>) or inforlinks (<https://www.infolinks.com/>) can be integrated to the website to display advertisement in website.

### **Express Resources**

* [ExpressJS](http://expressjs.com/)
* [Connect](https://github.com/senchalabs/connect)
* [Express Wiki](https://github.com/expressjs/express/wiki)
* [morgan](https://github.com/expressjs/morgan)
* [body-parser](https://github.com/expressjs/body-parser)

### **Other Resources**

* [Understanding Express.js](http://evanhahn.com/understanding-express/)
* [A short guide to Connect Middleware](https://stephensugden.com/middleware_guide/)

<p>

There are four main layers for a website, frond end, web service, business service and database. Frond end is for visualization and show user interface to customer. Frond end is the essential part for most of website, excepting providing API services. For the simple static website, front end could be the only part which is required. The other three are from back end. Web service is to accept frond end request and return related information/web page to frond end. The business service is to hold the business logic which could be one business service component or multiple business services which are always the case for complex web site. The last but not the least is done storage database. The database is to persist the webservice status, business data, user’s input etc.

</p>

<h3>Frond end</h3>

<p>

The main part for the frond end technology is html, css and java script. Duke university provide a great open course about html, css and java script. https://www.coursera.org/learn/duke-programming-web. W3school also provides tutorials about html, css, js etc. https://www.w3schools.com/. Codepen is a great tool to test webpage https://codepen.io/anon/pen/pxmZVM?&editors=0010. Dust (http://www.dustjs.com/) is one technology to provide html template. For this tutorial, we only use basic html, css and js with jQuery.

</p>

<h3>Web Service</h3>

<p>

There are many different technologies to build web service, such as php, jsp, asp, nodejs etc. Nodejs (https://nodejs.org ) is selected as it’s the newest technology for building web service. Kraken (http://krakenjs.com) is one framework that is based on nodejs. In this tutotial, express (https://expressjs.com/) and react (https://reactjs.org/docs/introducing-jsx.html) are used. Other resources

</p>

<p>https://stackoverflow.com/questions/47371904/e-unable-to-locate-package-npm </p>

<p>https://code.visualstudio.com/docs/nodejs/nodejs-tutorial.</p>

<h3>Business service</h3>

<p>

There are many ways to build business service. In this tutorial, we use java based spring boot technology to build restful service with json format.

</p>

<h3>Development environment</h3>

<p>

As webstorm is not free, visual studio code is best free web integrated development environment. Visual studio code is also for nodejs web service.

</p>

<p>

For the back-end java service, intelliJ is recommended IDE as eclipse is too slow and heavy.

</p>

<h3>Deployment environment.</h3>

<p>

After the code is done and tested locally, it’s ready to be deployed so the rest of the world could access it. There are several places the service could be deployed. The traditional one is to deploy to rent or buy one or several machine in IDC and deploy the components to it. You can also buy machines and put it in your home, and deploy the services to the machines. Any way you need apply a public IP, so other people could access it from any place with internet connection. Public cloud provides convenience and one stop solution for hosting services. Amazon aws and google cloud are top two public clouds that being used by difference purpose to host web/business services. The existed app engine can be used to deploy component or the components can be wrapped to docker and then deployed dockized component to Amazon aws or google cloud.

</p>

<p>

Some documents about deployment.

</p>

<p>Deploy: https://www.youtube.com/watch?v=n4svrNcAkJg</p>

<p>https://stackoverflow.com/questions/15444020/cannot-find-module-findup-sync-when-trying-to-run-grunt</p>

<p>https://cloud.google.com/storage/docs/hosting-static-website</p>

<p>https://github.com/jooyeong/kraken</p>

<p>Docker: https://medium.com/@cramirez92/build-a-nodejs-cinema-microservice-and-deploying-it-with-docker-part-1-7e28e25bfa8b</p>

<h3>Domain name</h3>

<p>

In order for the rest of the world access your webservice easier, domain name is required which is easy to remember rather than an ip address. I recommend to use google domain if you use google cloud, or Amazon router 35 if aws is used for web service. Otherwise godaddy is traditional web site to apply domain name. Use the same domain provider as the web service, will save you tons of time to config it.

</p>

<h3>Analytics</h3>

<p>

Google analytics (https://analytics.google.com/) is easy and free way to integrate for web site customer access analysis.

</p>

<h3>Marketing</h3>

<p>

Google search engine and bing search engine need to be added for SEO (Search engine optimization). Both search engines need site map, https://www.xml-sitemaps.com/ is a free and easy to use online site map generate tool.

</p>

<p>

Good adsense (https://www.google.com/adsense/start) or inforlinks (https://www.infolinks.com/) can be integrated to the website to display advertisement in website.

</p>