

UI 1.0 Toolkit

Ambika, Khushpreet, Soujanya and Madhavi

September 22, 2014

Contents

1	Introduction to UI Toolkit	1
1.1	Why is this needed?	2
2	Structure of the lab	2
3	The UI Template structure	5
4	How to use the template?	6
4.1	Case 1: Unstructured labs	6
4.1.1	Procedure for using scripts	7
4.2	Case 2: Labs hosted with Amrita Format (Collaborator Framework)	8
4.2.1	Necessary instructions for using scripts	8
4.2.2	Procedure for using scripts	9
4.2.3	Organize the downloaded content	10
4.2.4	Execute the script to transfer contents for a lab	10
4.2.5	Transfer the content into blue icon theme format . . .	11
4.2.6	Observe the UI transfer(mapping)	11

1 Introduction to UI Toolkit

The UI Toolkit consists of the UI 1.0 template of Virtual-labs and helper scripts that extracts content of labs hosted with Amrita UI format to new UI framework. The whole idea is to move the labs to new framework i.e Virtual-labs UI 1.0.

So basically a UI framework is software tools for building software programs to run on web. It is fully based on Web Standards such as HTML5, CSS and JavaScript, so that we can take an advantage of browser. A UI framework provides a set of CSS classes and/or Javascript functions that:

1. Simplifies the development of layout
2. Provides an attractive look and feel
3. Provides consistent results across browser with a lot of useful components.

With the basic web application toolkit we can write a design efficient labs with the framework that we have. The reason for going with html5 is because html5/css have some great new capabilities, they are just an evolution from html/css. The whole idea of building the framework is that it can be reused.

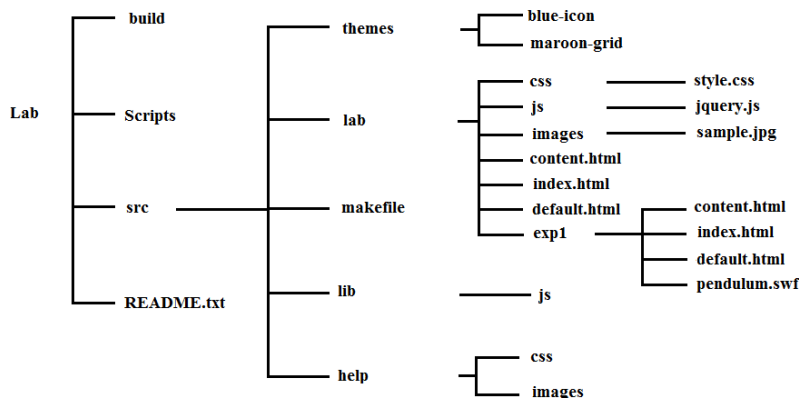
1.1 Why is this needed?

1. **Unified Look and Feel:** One is uniformity i.e uniform user interface for authoring and interacting with the virtual labs. Also, effective use of these labs depends on simple and uniform web interfaces and a homogeneous hosting framework.
2. **Coding standards:** Its necessary to follow good programming practices to make the code efficient and readable.
3. **Upgrading the widgets/components:** Template allows to modify/update components easy.
4. **Better versions of the template:** If better versions of the template is created in the future, the lab content can be automatically transferred.

2 Structure of the lab

Once your sources are ready with the content i.e (theory, procedures, experiments (simulations), quiz, images, videos etc). Before we move on to the procedure, it is essential to understand the structure of UI template 1.0.

Structure of Lab



This is how the lab should be structured.

The ui has src and scripts directory inside it. The ui/ directory has the following file structure:

`ui/src/lab/`

All the source files for the lab are here. The files `index.html` and `default.html` have been made read only as these need not be edited. In case you want to edit the html pages, you should edit only the `content.html` file in `ui/src/lab/` and the `ui/src/lab/exp1/` directory.

While adding new experiments, make sure you create new directories following the same structure under `ui/src/lab/"EXPERIMENT-ID"` and change the `content.html` in the same directory. Make sure inside the `ui/src/lab/"EXPERIMENT-ID"` directory, the files `index.html` and `default.html` are not edited. In addition, the `ui/src/lab` directory contains `js`, `images`, and `css` directories. These directories are there to hold your own custom files.

`ui/src/lib/`

All the library files (like MathJax, jQuery, modernizr) are in this directory.

`ui/src/themes/`

Two themes are supplied with this release.

- maroon-grid (uses iiith style, which does not display icons)

- blue-icon (uses the iitb style, which uses icons for sections)

Of course, you are free to design your own theme, and place it in the themes directory.

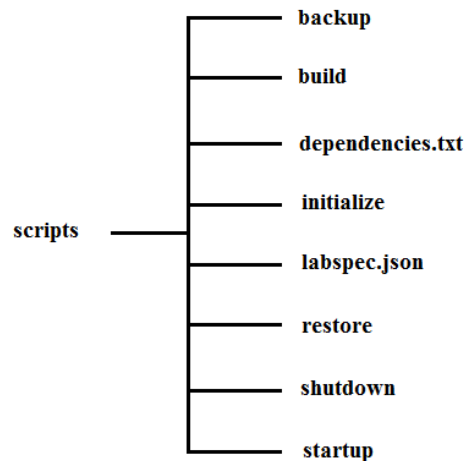
`ui/src/makefile`

This file is used for building a release.

`ui/scripts`

The scripts folder should contain json file(lab descriptor), shell scripts, dependencies file.

Files within scripts



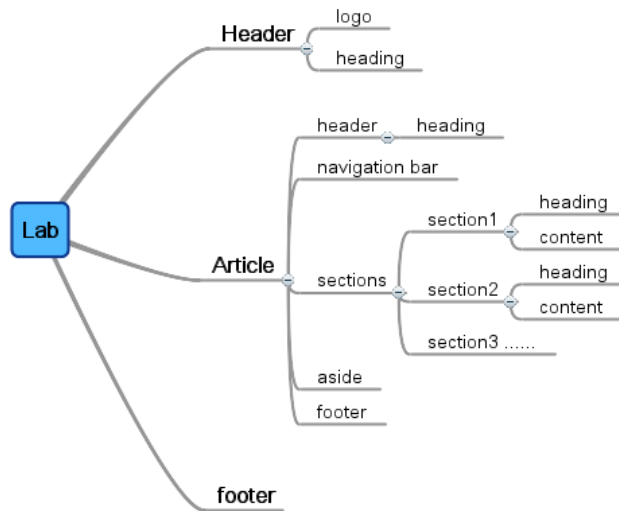
1. **Build:** Build Script runs the makefile and copies the build folder to localhost. The script is invoked by initialize script.
2. **Dependencies:** Mention all the server-side dependencies of the lab in dependencies.txt
3. **Initialize:** initialize script creates the initial environment for the Computer Programming.Lab by installing the server side dependencies for the lab and invokes the build script.To use initialize script, run the

command.initialize scripts/dependencies.txt.initialize script takes dependencies.txt as an argument and installs the packages mentioned in the dependencies.txt file.

4. **Backup:** Stores the backup of the database used by the lab.
5. **Restore:** Restore Script restores the backup of the database used by the lab.
6. **Shutdown:** shutdown script stops all the services on the container required to run the lab for the safe dissembling of the lab.
7. **Startup:** startup script starts all the services on the container required to run the lab.
8. **labspec.json:** This is an automated script for building the lab on container automatically.

3 The UI Template structure

All the labs should follow the general structure of the template.



Basically there will be three sections: Header, Article and footer.

Header generally contains the logo and heading of the page. **Article** contains the content of the page. It can be further divided into subsections

i.e header, footer, content etc. **Footer** contains licensing etc. Header and Footer remains same in all the webpages.

4 How to use the template?

There are two ways of lab conversion

4.1 Case 1: Unstructured labs

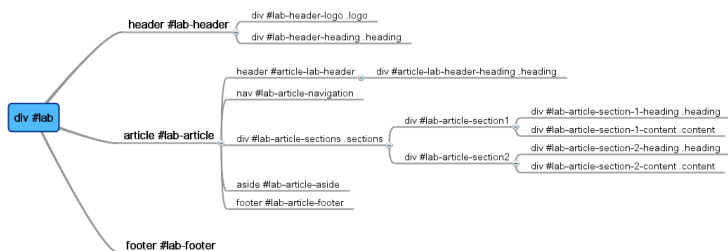
Editing content.html placed within lab folder

This template features a demo of a science lab with a sample “simple pendulum” experiment. The following section will help you to configure this template as per your lab experiments.

Mention the lab name under header and place the labs content under article followed by sections and order the articles accordingly. Also, list out the experiments in content.html. This is how the html page of the lab should be structured and we recommend to use these id’s for the lab.

The ui/src/lab/ directory contains three files:

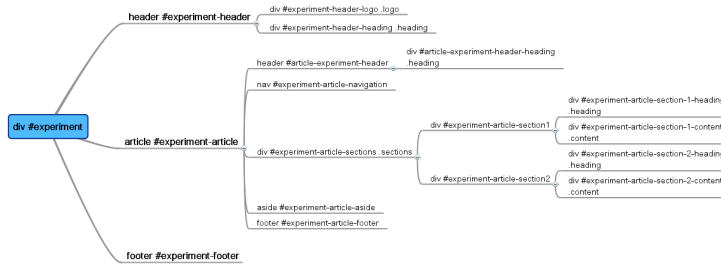
- index.html
- default.html
- content.html



Editing content.html placed within each experiment folder

The same structure has been followed inside the ui/src/lab/”EXPERIMENT_ID” directory. The description and the read-write permits for those file have been tabulated here as follows :

file	permissions	description
index.html	read-only	home-page
default.html	read-only	default-content
content.html	read-write	editable-content



This is how the html page of the experiment should be structured and we recommend to use these ids for the experiments.

4.1.1 Procedure for using scripts

1. Once the content is added, then edit the “dependencies.txt” file and list out all the dependencies of your lab.
2. Based on the operating system scripts will differ, so make sure which scripts you are using. If your operating system is “centos” then use “centos-template” scripts otherwise use “ubuntu-template” scripts.
3. Copy the scripts from desired template (i.e. ubuntu-template or centos-template) and paste inside your template folder.
4. Now open the terminal and change your directory to template and then scripts.
5. Run the following commands

```

chmod 777 initialise.sh
./initialise.sh

```

This script will automatically install all the dependencies and will run the “make” file in order to generate a build folder.

Note: if you come across any error while running the script, you need to login as root user and follow the same procedure.

```

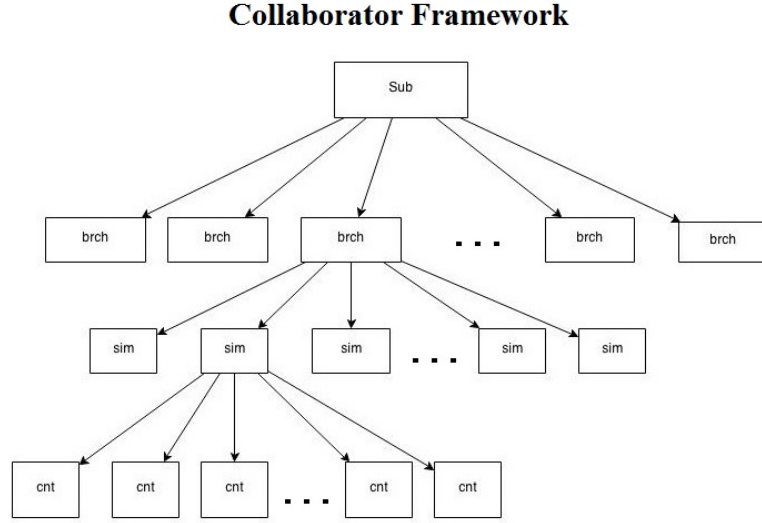
sudo su
or
su -

```

4.2 Case 2: Labs hosted with Amrita Format (Collaborator Framework)

For converting the labs from Amrita format to UI 1.0, we have scripts in the UI toolkit that automates the process to a large extent but there are some changes that need to be done manually.

Here is the framework that Amrita labs are designed.



Format.jpg

4.2.1 Necessary instructions for using scripts

Let me start with an example, say for the lab chosen at a random is given by the following url: (<http://iitkgp.vlab.co.in/?sub=39&brch=124&sim=922&cnt=1>) Here the breakup of the url is as per the following scheme, firstly the labs are classified under subjects or domains indicated by “sub” and then each branch or lab under those domains is indicated by a separte “brach” number. After that each experiment in the lab is indicated by a simulator number named in the url as “sim”. Lastly the count or “cnt” number indicates the breakup of pages under each experiment, namely theory, introduction etc.

As per the given url in the example, we have **sub=39**, as the “*Electrical Engineering*” domain and **brch=124**, as the “*analog signals, network and measurement laboratory*”, finally the **sim=922**, refers to the experiment named “*Verification of Superposition Theorem*” and **cnt=1**, indicated the “*Theory*” page under it.

- **organise.sh**

organise.sh is a shell script to get the contents from the lab url and store them in an organized format as university->subject->branch->labs->experiments.

Now the structure of the lab is analyzed and the script breaks up the url accordingly and creates the folders recursively in an organized format.

- **scrapper.py**

scrapper.py is a python based script to transfer the contents under one lab-experiment in an automated way to the requisite UI format, which is the new Blue-icon theme format.

A Python library called “Beautiful Soup”, designed to parse xml, html5 files and tree traversals has been used here to parse the contents of each lab-experiment’s html files. These parsed contents are mapped to the Blue-icon theme format and copied into the “content.html” using python commands.

Ref: <http://www.crummy.com/software/BeautifulSoup/>

- **link_changer.sh**

link_changer.sh is also a shell script to link the images downloaded in the userfiles folder to contents.html of the lab-experiments mapped using the former scheme.

Here the script replaces the path of the images inside each content.html file for all experiments in the lab with the appropriate and desired path of the new directory after the mapping is done.

- **all.sh**

It will automatically run both scrapper.py and link_changer.sh. It also builds the lab automatically.

Note: Follow the same procedure for every experiment. After converting all the experiments, we need to rename the “exp1” folder to experiment-wise, e.g: exp1, exp2, exp3 so on and place them inside lab folder.

4.2.2 Procedure for using scripts

The following instructions need to be followed for successfully porting the content from Amrita model theme to blue icon theme.

- **Check out the scripts from github**

Check out the lik (lab integration kit) from github

```
git clone https://github.com/Virtual-Labs/lab-integration-kit.git
```

Find the scripts from following path:

Go to: lab-integration-kit-> ui-toolkit-1.0->scripts/

- **Download and store the content**

Create a folder called “map”

```
> mkdir map
```

- **Move into the folder map**

```
> cd map
```

- **Getting the content from desired url**

Run the wget command under the some directory.

```
> wget -r -e robots=off http://iitkgp.vlabs.co.in
```

Note: It is necessary to override robots.txt file to download the images along with the other contents of the lab.

4.2.3 Organize the downloaded content

Run the organise .sh file and prior to that give it execute permission.

```
> chmod 777 organise.sh  
> ./organize.sh
```

4.2.4 Execute the script to transfer contents for a lab

Given a lab we have to run the code manually for each page and more precisely for all the “cnt” number in our case under each experiment(sim). The command to do it articulately are given in following subsections: Go inside the subject->branch->lab->experiment use cd command followed by path name.

4.2.5 Transfer the content into blue icon theme format

“scrapper.py” is used to transfer the content from Amrita’s format to UI 1.0. “link_changer.sh” is used to update the links to the images inside the experiment. “make” is used to build the lab.

run the following command:

```
> ./all.py
```

Note: if any error comes while running this script then become a root user and follow the same procedure.

```
sudo su
```

```
or
```

```
su -
```

Note: all.py will automatically runs “scrapper.py”, “link_changer.sh” and “make”. It will open in the browser with Simple pendulum sample experiment. So, click on that link. Mostly the template would be ready with the content added, but the images at each tab might be missing. It must be something like this:



Note: In that case copy the images from the /template/src/lab/images and place it in the same path that the experiment follows.

4.2.6 Observe the UI transfer(mapping)

This is how the lab structured after conversion.

[Theory](#)
[Procedure](#)
[Labvw calc](#)
[Pre expt quiz](#)
[Video](#)
[Download](#)
[Simulator](#)
[Post expt quiz](#)
[Help](#)
[Feedback](#)

PROCEDURE

1. Cleaning the Setup : The setup is cleaned to start a new Experiment.
2. Mixing in given ratios : A mixture of benzene and toluene is prepared and put into the flask
3. Power is Supplied : Then the power is supplied and the mixture begins to boil.
4. Equilibrium is Reached: After some time equilibrium is reached, this is marked by steady temperature in both the phases.
5. Measuring RI : Samples of condensed vapor and liquid are taken. To calculate the composition we need to measure the refractive index of the sample and look into the chart.

Waiting for ahs.amrita.ac.in...