Wireless LANs

|  |
| --- |
| IEEE 802.11 Timeline & Ballot Tool |
| Date: 2010-11-10 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Alex Ashley | NDS Ltd | One London Road, Staines, Middlesex, TW18 4EX, UK |  | aashley at nds dot com |
|  |  |  |  |  |

Abstract

The IEEE 802.11 timeline tool is a web application that is currently being hosted by the Google App Engine.

The application is based upon the model, view, controller paradigm. It uses databases to hold information on the IEEE 802.11 projects and ballots, scripts to control the “business logic” of viewing and modifying this data, and HTML templates that render the data that you see in your web browser.

The main advantage of this approach is that it allows changes to the visual appearance without requiring any change to the data or control logic.

## Introduction

The IEEE 802.11 timeline tool is a web application that is currently being hosted by the Google App Engine. The URL for the site is:

http://ieee80211timeline.appspot.com/

To ensure that this application stays within the free usage quota, access to all parts of this website apart from the front page requires a username and password. Contact Alex Ashley if you would like to request access.

The application is based upon the model, view, controller paradigm. It uses databases to hold information on the IEEE 802.11 projects and ballots, scripts to control the “business logic” of viewing and modifying this data, and HTML templates that render the data that you see in your web browser.

The main advantage of this approach is that it allows changes to the visual appearance without requiring any change to the data or control logic.

## Timeline Tool

The timeline tool is used to view, manage and export the IEEE 802.11 timeline chart.



### Edit/Delete Project

To modify a project, click on the link in the first column (“IEEE Project and Final Document”) of the table, for the project you are editing. For example, to edit the 11aa project, you click on the “IEEE Std P802.11aa” link in the left hand column. This will take you a web based form that contains all the data for the selected project:



You simply edit each field and then click the “Save” button to save changes. The “Cancel” button will cause all changes to be lost and return you to the previous page. The “Delete” button will take you to a “Do you really want to delete this project?” page, where you need to confirm that you really do want to delete this project.

The date fields must be entered in the form YYYY-MM-DD (e.g. 2010-06-21).

The URL fields (“Par”, “Task group url”) must not have any spaces in their URL. If the document you need to point to has spaces in them, replace each space with %20. For example “my document.doc” becomes “my%20document.doc”.

The “Baseline” field is used for two different purposes. For an amendment project, it points to the baseline standard document on which this amendment is based. For example the baseline for P802.11v is “802.11-2007” where as the baseline for P802.11aa is 802.11 (REVmb).

For a revision standard (e.g. REVmb) the “Baseline” field is used to point to the standard that it replaces. For REVmb, this means that the “Baseline” field is set to “P802.11-2007”.

The “Order” field also has two different purposes. For an amendment project, it is set to the amendment number, with 1 as the first amendment. For a revision standard, it is set to a number greater than the amendment number that is incorporated in that revision. For example if the REVmb project has its order field set to 10, it means that it will incorporate amendments 1 to 9.

### Add Project

There is a menu at the top of the page:



Clicking on the “Add Project” button takes you to a page that is identical to the edit project page, except that it allows a new project to be created.

### Edit Ballot

Each ballot shown on the timeline page is a clickable link that will take you to a page to edit the ballot.



This will take you to a web form that allows editing of the data for the selected ballot:



You simply edit each field and then click the “Save” button to save changes. The “Cancel” button will cause all changes to be lost and return you to the previous page. The “Delete” button will take you to a “Do you really want to delete this ballot?” page, where you need to confirm that you really do want to delete this ballot.

The date fields should be entered in the form YYYY-MM-DD (e.g. 2010-06-21).

The URL fields must not have any spaces in their URL. If the document you need to point to has spaces in them, replace each space with %20. For example “my document.doc” becomes “my%20document.doc”.

Sponsor ballots do not have a working group member visible number, but each ballot in the database needs a ballot number because it is used as the primary key in the database. When the existing sponsor ballot data was imported in to the database, sponsor ballots were given ascending numbers starting at 10000.

### Add Ballot

Clicking on the “Add Ballot” button takes you to a page that is identical to the edit ballot page, except that it allows a new ballot to be created.

The “Number” field is pre-populated with a unique number that can be used for adding a sponsor ballot.

### Processing Updates

In order to allow the timeline page to render in a timely fashion, behind the scenes some extra data is pre-computed and stored in the database. Background tasks are triggered to run on the server hosting the timeline tool as soon as updates are made to projects or ballots. Sometimes you will see the title at the top of the page will change to “Processing Updates” and the export menu will be disabled. When you see “processing updates” you can still interact with the web page, but some changes might not have propagated in to the page you are viewing. The page will be automatically reloaded at regular intervals until all the background process has completed.

### Export Timeline

The “Export HTML” and “Export SHTML” create “clean” versions of the timeline page, without the top level menu and without the hyperlinks to edit ballots and projects. The “Export HTML” version will produce a standalone web page that can be uploaded to the “grouper” HTTP server. The “Export SHTML” server produces a file that contains SHTML server side include directives that cause the menu template to be incorporated.

## Letter Ballot tool

The Letter Ballot tool allows production of the Working Group and Sponsor Ballot pages that list the open and closed ballots. This tool has two links: “Working Group Ballots” and “Sponsor Ballots”

### Working Group Ballots

The “Working Group Ballots” link takes you to a page that shows the open and closed working group ballots, and allows ballots to be added, edited and deleted.



Each item in the “LB” column has a link that takes you to a page that allows the ballot to be edited. The details of the edit ballot page are identical to the timeline tool “edit ballot” page.

Each item in the “Group(s)” column has a link that takes you to a page that allows a project to be edited. The details of the edit project page are identical to the timeline tool “edit project” page.

### Sponsor Ballots

The “Sponsor Ballots” link takes you to a page that shows the open and closed sponsor ballots, and allows ballots to be added, edited and deleted.



Each item in the “Group” column has a link that takes you to a page that allows a project to be edited. The details of the edit project page are identical to the timeline tool “edit project” page.

Each item in the “Draft” column has a link that takes you to a page that allows the ballot to be edited. The details of the edit ballot page are identical to the timeline tool “edit ballot” page.

### Processing Updates

In order to allow the ballot pages to render in a timely fashion, behind the scenes some extra data is pre-computed and stored in the database. Background tasks are triggered to run on the server hosting the timeline tool as soon as updates are made to projects or ballots. Sometimes you will see the title at the top of the page will change to “Processing Updates” and the export menu will be disabled. When you see “processing updates” you can still interact with the web page, but some changes might not have propagated in to the page you are viewing. The page will be automatically reloaded at regular intervals until all the background process has completed.

## Exporting Data

The project and ballot data can be exported from the database, as a way of keeping a backup of this data and to allow bulk edits of the data without using the web based tool.

The format of the data is a comma separated value file (CSV) that can be widely parsed by many applications, for example Microsoft Excel.

The file consists of two sections, the project section and the ballot section.

The project section consists of a header of:

pk,name,description,doc\_type,par,task\_group,task\_group\_url,doc\_format,doc\_version,baseline,order,par\_date,par\_expiry,initial\_wg\_ballot,recirc\_wg\_ballot,sb\_form\_date,sb\_formed,initial\_sb\_ballot,recirc\_sb\_ballot,mec\_date,mec\_completed,wg\_approval\_date,ec\_approval\_date,revcom\_approval\_date,ansi\_approval\_date,withdrawn\_date,withdrawn,history,wg\_approved,ec\_approved,published,slug

This header is followed by each project, one line per project.

The ballot section consists of a header of:

number,draft,opened,closed,ballot\_type,pool,vote\_for,vote\_against,vote\_abstain,vote\_invalid,comments,instructions\_url,draft\_url,redline\_url,resolution\_url,template\_url,pool\_url,project\_id,project.task\_group

This header is followed by each ballot, one line per ballot.

The format and contents of each field is described in the appendix of this document.

## Importing Data

The import page allows a previously exported CSV file to be loaded in to the database. If the primary key field refers to an item that is already in the database, the item in the database will be updated. If the primary key refers to an item that is not in the database, the item will be added. The one exception to this rule is that ballots may have a ballot number of zero, which causes the import tool to auto-create a number for this ballot.

When importing a ballot, the project\_id field must either contain the primary key of the project to which the ballot pertains, or has the value zero. If the project\_id is zero, the project.task\_group field must contain the exact text of the task\_group field of the project. For example project\_id=0 and project.task\_group=TGmb.

The import tool also allows an HTML version of the working group or sponsor ballot pages to be directly imported. The import tool will parse the HTML to extract the HTML tables and try to parse ballots from these pages. This feature is obviously very sensitive to changes in the HTML layout and should be considered a second best method of importing, compared to using a CSV file.

## Application Environment

The application is written in the Python programming language and is based upon the Django web development framework. As the application is hosted on the Google App Engine (GAE), a database abstraction layer is required because Django requires the use of a relational database, but the GAE datastore is non-relational database. The timeline tool uses the django-nonrel abstraction layer to allow Django to use the GAE datastore.

<http://www.python.org/>

<http://code.google.com/appengine/>

<http://www.djangoproject.com/>

<http://www.allbuttonspressed.com/projects/django-nonrel>

To run the timeline tool locally, you need to install Python, the GAE development environment, unzip the timeline tool source code and then place the django-nonrel source code in the same directory as the timeline tool source code, as described below.

Download the following:

* [Google App Engine for Python](http://code.google.com/appengine/downloads.html#Google_App_Engine_SDK_for_Python)
* [django-nonrel](http://bitbucket.org/wkornewald/django-nonrel/get/tip.zip)
* [djangoappengine](http://bitbucket.org/wkornewald/djangoappengine/get/tip.zip)
* [djangotoolbox](http://bitbucket.org/wkornewald/djangotoolbox/get/tip.zip)

Install the Google App Engine. Unzip the django-nonrel source code in to a temporary location.

Copy the following folders into the same directory where you have unzipped the source code:

* django-nonrel/django => <timeline\_dir>/django
* djangotoolbox/djangotoolbox => <timeline\_dir>/djangotoolbox
* django-dbindexer/dbindexer => <timeline\_dir>/dbindexer
* djangoappengine => <timeline\_dir>/djangoappengine

Once finished, you should have a directory that contains:

<DIR> ballot

<DIR> django

<DIR> djangoappengine

<DIR> djangotoolbox

<DIR> media

<DIR> project

<DIR> templates

<DIR> timeline

<DIR> util

app.yaml

cron.yaml

index.yaml

LICENSE-2.0.txt

manage.py

queue.yaml

robots.txt

settings.py

test.bat

urls.py

\_\_init\_\_.py

To test that the code is working correctly, run the unit tests:

If using Windows, by typing the following in a command prompt window:

runtests.bat

If you are running the software on a Unix platform, use:

manage.py test util ballot project timeline

This should produce “OK” after running all the tests.

The local development web server can be started using:

manage.py runserver

and then opening <http://localhost:8000/> in a web browser.

# Appendix: Database model

|  |
| --- |
| **Project Model** |
| **Field Name** | **Type** | **Maximum Size** | **Required** | **Description** |
| pk | IntegerField |  | Yes(primary key) | Primary key. Not editable. |
| name | CharField | 20 | Yes | Name of standard/amendment (e.g. 802.11aa), WITHOUT year designation |
| slug | SlugField | 15 | N/A | Used to create a unique name for each group. Not editable. |
| description | CharField | 100 | Yes | Project Description |
| doc\_type | CharField enum | 4 | Yes | ( 'A', 'Amendment'),( 'STD', 'Standard'),( 'RP', 'Recommended Practice'),( 'COR' , 'Corrigendum') |
| par | URLField |  | No | URL pointing to PAR document |
| task\_group | CharField | 10 | Yes | Name of task group (TG..) |
| task\_group\_url | URLField |  | No | URL pointing to TG status page |
| doc\_format | CharField | 20 | No |  |
| doc\_version | DecimalField | ####.## | No | Version number of current draft |
| baseline | IntegerField |  | No | Primary key value of baseline standard. |
| order | IntegerField |  | Yes | Used to set amendment ordering, with lowest number first. For a revision standard, order must be larger than any included amendment |
| par\_date | DateField |  | Yes | Date that PAR was first approved |
| par\_expiry | DateField |  | No | Date that PAR will expire |
| history | CharField | 60 | No | Dates when PAR was renewed (Must be in the form YYYY-MM-DD, YYYY-MM-DD, ..) |
| initial\_wg\_ballot | DateField |  | No | Estimated Date of initial WG ballot |
| recirc\_wg\_ballot | DateField |  | No | Estimated Date of recirculation WG ballot |
| sb\_form\_date | DateField |  | No | Date that sponsor ballot will be formed or was formed |
| sb\_formed | BooleanField |  |  | Sponsor pool has been formed |
| initial\_sb\_ballot | DateField |  | No | Estimated Date of initial SB ballot |
| recirc\_sb\_ballot | DateField |  | No | Estimated Date of recirculation SB ballot |
| mec\_date | DateField |  | No | Estimated/Actual date of mandatory editorial co-ordination |
| mec\_completed | BooleanField |  |  | MEC has been performed |
| wg\_approval\_date | DateField |  | No | Estimated/Actual date of WG approval |
| wg\_approved | BooleanField |  |  | WG has approved doc |
| ec\_approval\_date | DateField |  | No | Estimated/Actual date of EC approval |
| ec\_approved | BooleanField |  |  | EC has approved doc |
| revcom\_approval\_date | DateField |  | No | Estimated/Actual date of Revcom approval |
| published | BooleanField |  |  | Document has been published |
| ansi\_approval\_date | DateField |  | No | Actual date of ANSI approval |
| withdrawn\_date | DateField |  | No | Actual date when document was withdrawn |
| withdrawn | BooleanField |  |  | Document has been withdrawn |

|  |
| --- |
| **Ballot Model** |
| **Field Name** | **Type** | **Maximum Size** | **Required** | **Description** |
| number | IntegerField |  | Yes(primary key) | Number of this ballot. For sponsor ballots, you need to invent a number |
| draft | DecimalField | ####.## | Yes | The version number of the balloted draft |
| opened | DateField |  | Yes |  |
| closed | DateField |  | Yes |  |
| ballot\_type | CharField enum | 5 | Yes |  |
| pool |  |  | No | Number of voters in ballot pool |
| vote\_for | IntegerField |  | No |  |
| vote\_against | IntegerField |  | No |  |
| vote\_abstain | IntegerField |  | No |  |
| vote\_invalid | IntegerField |  | No | Number of NO votes without comments |
| comments | IntegerField |  | No | Number of comments |
| instructions\_url | URLField |  | No | URL pointing to ballot instruction |
| draft\_url | URLField |  | No | URL pointing to draft document |
| redline\_url | URLField |  | No | URL pointing to redline version of draft |
| instructions\_url | URLField |  | No | URL pointing to ballot instruction |
| resolution\_url | URLField |  | No | URL pointing to comment resolutions document |
| template\_url | URLField |  | No | URL pointing to comment submission template |
| pool\_url | URLField |  | No | URL pointing to voter pool |

 **References:**