django-timepiece Documentation

Release 0.5.4

October 04, 2012

CONTENTS

django-timepiece is a multi-user application for tracking people's time on projects. Complete documentation is available on Read The Docs.

CHAPTER

ONE

FEATURES

- A simple CRM with projects and businesses
- User dashboards with budgeted hours based on project contracts
- Time sheets with daily, weekly, and monthly summaries
- Verified, approved, and invoiced time sheet workflows
- Monthly payroll reporting with overtime, paid leave, and vacation summaries
- Project invoicing with hourly summaries

CHAPTER

TWO

REQUIREMENTS

django-timepiece is compatible with Python 2.{6,7}, Django 1.{3,4}, and PostgreSQL. PostgreSQL is the only offically supported database backend and, therefore, requires psycopg2. django-timepiece also depends on the following Django apps:

- python-dateutil
- django-selectable
- django-pagination
- django-compressor
- django-bootstrap-toolkit

django-timepiece uses Sphinx and RST for documentation. You can use Sphinx to build the documentation:

- docutils
- Sphinx

A makefile is included with the documentation so you can run *make html* in the *doc/* directory to build the documentation.

CHAPTER

THREE

INSTALLATION

1. django-timepiece is available on PyPI, so the easiest way to install it is to use pip:

   ```
   $ pip install django-timepiece
   ```

2. Ensure that less is installed on your machine:

   ```
   # Install node.js and npm:
   $ sudo apt-get install python-software-properties
   $ sudo add-apt-repository ppa:chris-lea/node.js
   $ sudo apt-get update
   $ sudo apt-get install nodejs npm

   # Use npm to install less:
   $ npm install less -g
   ```

3. Add *timepiece* and its dependencies to `INSTALLED_APPS` in *settings.py* and run syncdb:

   ```
   INSTALLED_APPS = (
       ...
       'bootstrap_toolkit',
       'compressor',
       'pagination',
       'selectable',
       'timepiece',
       ...
   )
   ```

4. Add *django.core.context_processors.request* and django-timepiece context processors to `TEMPLATE_CONTEXT_PROCESSORS`:

   ```
   TEMPLATE_CONTEXT_PROCESSORS = (
       "django.contrib.auth.context_processors.auth",
       "django.core.context_processors.debug",
       "django.core.context_processors.i18n",
       "django.core.context_processors.media",
       "django.contrib.messages.context_processors.messages",
       "django.core.context_processors.request",       # <----
       "timepiece.context_processors.extra_nav",       # <----
       "timepiece.context_processors.active_entries",  # <----
   )
   ```

5. Configure compressor:

   ```
   COMPRESS_PRECOMPILERS = (
       ('text/less', 'lessc {infile} {outfile}'),
   ```

```
)
COMPRESS_ROOT = '%s/static/' % PROJECT_PATH
```

6. Add URLs for django-timepiece and selectable to *urls.py*, e.g.:

```
urlpatterns = patterns('',
    ...
    (r'^selectable/', include('selectable.urls')),
    (r'', include('timepiece.urls')),
    ...
)
```

7. Add the `django.contrib.auth` URLs to *urls.py*, e.g.:

```
urlpatterns = patterns('',
    ...
    url(r'^accounts/login/$', 'django.contrib.auth.views.login',
        name='auth_login'),
    url(r'^accounts/logout/$', 'django.contrib.auth.views.logout_then_login',
        name='auth_logout'),
    url(r'^accounts/password-change/$',
        'django.contrib.auth.views.password_change',
        name='change_password'),
    url(r'^accounts/password-change/done/$',
        'django.contrib.auth.views.password_change_done'),
    url(r'^accounts/password-reset/$',
        'django.contrib.auth.views.password_reset',
        name='reset_password'),
    url(r'^accounts/password-reset/done/$',
        'django.contrib.auth.views.password_reset_done'),
    url(r'^accounts/reset/(?P<uidb36>[0-9A-Za-z]+)-(?P<token>.+)/$',
        'django.contrib.auth.views.password_reset_confirm'),
    url(r'^accounts/reset/done/$',
        'django.contrib.auth.views.password_reset_complete'),
    ...
)
```

8. Create registration templates. For examples, see the registration templates in *example_project/templates/registration*. Ensure that your project's template directory is added to `TEMPLATE_DIRS`:

```
TEMPLATE_DIRS = (
    ...
    '%s/templates' % PROJECT_PATH,
    ...
)
```

Development sponsored by Caktus Consulting Group, LLC.

Contents:

3.1 Settings

3.1.1 EXTRA_NAV

This is an optional settings that is used to add extra navigation items to the global navigation bar displayed at the top of each django-timepiece page. The setting uses Twitter Bootstrap's dropdown navigation. The setting can be declared as follows:

```
EXTRA_NAV = {
    'Dropdown Text': [
        ('named_url_1', 'Dropdown Item #1',),
        ('named_url_2', 'Dropdown Item #2',),
    ], ...
}
```

For example, the following would add a "Paste Bin" dropdown menu with the two links "Make a Paste" and "View my Pastes" as menu items in that dropdown.

```
EXTRA_NAV = {
    'Paste Bin': [
        ('create_paste_view', 'Make a Paste',),
        ('user_paste_view', 'View my Pastes',),
    ],
}
```

You can add as many menu items as you wish and are only limited by the space available in the navigation bar. Any navigation item that you add will appear following the django-timepiece related navigation items.

Note: You must have the context processor *timepiece.context_processors.extra_nav* listed in Django's *TEMPLATE_CONTEXT_PROCESSORS* setting.

3.2 Testing

django-timepiece includes several different alternatives for testing. Test can be run using the default django test runner, through Tox, or with django-jenkins. Tox and django-jenkins are not required to run the tests for django-timepiece, but it is possible to use them:

```
pip install --upgrade tox django-jenkins
```

A Python module, `run_tests.py`, is included if you do not want to run tests using Tox. This is the Python module used to run tests when executing `python setup.py test`. The tests are run through Django, using Django's default test runner. It accepts an optional argument, `run_tests.py jenkins`, that runs the tests using django-jenkins. Running the tests with django-jenkins also requires you to install coverage and pep8.

django-timepiece inclues a Tox configuration file to run tests in a variety of environments:

- *py26-1.3* - Test using Python 2.6 and Django 1.3.x
- *py26-1.4* - Test using Python 2.6 and Django 1.4.x
- *py27-1.3* - Test using Python 2.7 and Django 1.3.x
- *py27-1.4* - Test using Python 2.7 and Django 1.4.x

You can run any of the environments listed above using: `tox -e name`. The tests are run through Django's default test runner, but you can also run the tests using django-jenkins along with tox by providing an extra argument: `tox -e name -- jenkins`.

3.3 Release Notes

3.3.1 0.6.0 (Released 10-04-2012)

- Updated version requirement for South to 0.7.6

- Updated version requirement for django-bootstrap-toolkit to 2.5.6
- Use Javascript to manage date filter links on Reports pages
- Use “empty” text when there is no Billable Report data to visualize
- Include auth groups select to Person creation form
- Added pagination and search to Previous Invoices page
- Show current project name and activity on Clock Out page
- Maintain selected month on link to Person time sheet from Payroll Report page
- Maintain selected month on link to Project time sheet from Outstanding Hours page
- Fixed division-by-0 bug on ContractAssignment admin page
- Fixed infinite loop when ordering by Project on ProjectContract admin page
- Prevent admin from requiring that all ProjectContract inlines be completed on Project creation
- Use default options for the filter form on the Hourly Report page

We also completed a full audit of the code, in which we deleted stale parts, removed unmaintained features, and made some simple cleanups:

- Migrated the `PersonSchedule.hours_per_week` field to the UserProfile model
- Deleted the AssignmentAllocation and PersonSchedule models
- Removed all projection-related code, including admin and model hooks, forms, views, templates, and *projection.py*
- Deleted *widgets.py*
- Removed unused fields from DateForm
- Removed unused templates and static files
- Removed unused utilities, template tags, and forms
- Cleaned up imports, used the `render` shortcut in all views, and used the new-style url in all templates
- Refreshed the example project and added missing templates and JavaScript files

3.3.2 0.5.4 (Released 09-13-2012)

- Projects on Invoices/Outstanding Hours page are sorted by status and then by name
- Weekly Project Hours chart uses horizontal zebra striping
- New permission added for approving timesheets
- Fixed a bug in Project Hours edit view that prevented deletion of multiple entries at once
- Added links to Person timesheet from Payroll Report page
- Added links to Project timesheet on Invoice page

3.3.3 0.5.3 (Released 08-10-2012)

- Added a “Billable Hours” report, which displays a chart of billable and non-billable hours for a selected group of people, activities, project types and date range.
- Improved usability of the payroll report

- Made forms with date ranges more consistent and DRY
- Added a restriction that prevents users from adding entries to months with approved or invoiced entries.
- Removed the link to edit weekly project hours for users without that permission
- Improved readability of report tables by changing the hover color to something more distinctive.

3.3.4 0.5.2 (Released 08-01-2012)

- Added "Project Hours" views, which allow managers to assign project hours to users in a spreadsheet-like interface.
- Simplified implementation of timezone support.
- Fixed a bug that was preventing the weekly totals in "Hourly Summary" of "My Ledger" from being displayed.
- Removed the display of "hours out of" in the "billable time" section of "My Work This Week" and added it to the "total time this week" section.

3.3.5 0.5.1 (Released 07-20-2012)

- Added compatability with Django 1.4 and timezone support
- Added mobile support for the dashboard (clocking in/out, ledger, etc.)
- Fixed a bug where the last billable day was calculated incorrectly
- Payroll report now lists types of projects under billable and non-billable columns
- Moved the "Others Are Working On" table to a popover in the navigation
- Work total table now includes the active entry
- Comment field available when clocking in to a project
- Added support for custom navigation through EXTRA_NAV setting
- Across the board styling changes

3.3.6 0.5.0 (Released 07-12-2012)

- Complete styling upgrade using Twitter Bootstrap
- Fixed permissions for client users that can't clock in
- Replaced deprecated message_set calls with new messages API calls
- Added django-bootstrap-toolkit requirement
- Included the top navigation bar inside of the app's templates.
- Made the project edit form use selectables for searching for businesses.
- Improved tox configuration of test database names
- Added a makefile and /docs for building documentation with Sphinx

3.3.7 0.4.2 (Released 06-15-2012)

- Fixed permissions for creating businesses.
- Hourly reports in “My Ledger” display previous weeks of the month if an overlapping entry exists.
- Fixed permissions for rejecting verified entries.
- Fixed a bug where you could verify entries while still clocked in.
- Added user selection for payroll reviewers to switch between timesheets.
- Fixed bug where the incorrect email was shown in the header.

3.3.8 0.4.1 (Released 06-04-2012)

- Made projects’ tracker URL’s appear on the project detail view.
- Added reasonable limits to the total time and pause length of entries.
- Users can now comment on the active entry while clocking into a new one.
- Fixed a bug with entries overlapping when clocking in while another entry is active.
- Added the ability for payroll reviewers to reject an entry, which marks it as unverified.
- Added a weekly total on the dashboard for all hours worked.
- The hourly summary in “My Ledger” now shows the entire first week of the month.
- Made payroll links to timesheets maintain the proper month and year.
- Made URL’s in entry comments display as HTML links
- Fixed permissions checking for payroll and entry summary views.
- Made project list page filterable by project status.
- Replaced django-ajax-select with latest version of django-selectable
- Added migration to remove tables related to django-crm

3.3.9 0.4.0 (Released 04-27-2012)

- Improved personnel timesheets with a simplified, tabbed layout.
- Improved efficency and consistency of entry queries
- Removed BillingWindow, RepeatPeriod, and PersonRepeatPeriod models, tables and related code.
- Removed the update billing windows management command as it is no longer needed.

3.3.10 0.3.8 (Released 02-16-2012)

- Converted invoice reference to a CharField for more flexibility
- Added list and detail views for project contracts
- Hour groups now show totals for each activity nested within them
- Moved unapproved and unverified entry warnings to the payroll summary page.

3.3.11 0.3.7 (Released 02-01-2012)

- Make create invoice page inclusive of date

3.3.12 0.3.6 (Released 02-01-2012)

- Allowed entries to be added in the future.
- Added per project activity restrictions.
- Allowed marking entries as ‘not invoiced’ and grouped entries together after clicking on “Mark as invoiced”
- Added the ability to view previous invoices and export them as csv’s
- Added the ability to group different activities together into Hour Groups for summarizing in invoices.

3.3.13 0.3.5 (Released 12-09-2011)

- Optimized Payroll Summary with reusable code from Hourly Reports.
- Removed use of Textile and used the linebreaks filter tag in its place.

3.3.14 0.3.4 (Released 11-14-2011)

- Added a new Hourly Reports view with project hours filtered and grouped by user specified criteria.
- Hourly Reports, General Ledger and Payroll Summary are now subheadings under Reports.
- Improved My Ledger with row highlighting, better CSS and a title attribute.
- Fixed Invoice projects to return the date range with m/d/Y.

3.3.15 0.3.3 (Released 10-31-2011)

- Fixed Time Detail This Week on Dashboard to show correct totals
- Fixed Billable Summary on My Ledger to show totals for unverified hours

3.3.16 0.3.2 (Released 10-28-2011)

- My Active Entries on Dashboard now shows the hours worked thus far
- Improved My Ledger by adding a comments column and a redirect from the edit entry link
- Fixed issues related to the hourly summary option not appearing for some users
- Fixed issues with date accuracy in weekly headings on ledger pages
- General ledger now sorts users by last name
- Enhanced project time sheets with an activity column and a summary of hours spent on each activity.
- Invoice projects page now shows project status
- Activity on clock in page now defaults to the last activity clocked on that project
- Payroll report only shows users that have clocked hours for the period.

3.3.17 0.3.1 (Released 10-20-2011)

- Moved to GitHub (and git)
- Add hourly summary page to report daily, weekly, and monthly hours
- Refactored weekly overtime calculations to use ISO 8601

3.3.18 0.3.0 (Released 10-03-2011)

- Removed ability to maintain multiple active entries
- Enhanced logic on clock in and add entry pages to check for overlapping entries
- Fixed date redirect when marking projects as invoiced
- Fixed issues related to the “Approve Timesheet” link missing
- Include billable, non-billable, uninvoiced, and invoiced summaries on person timesheet
- Use select_related in a few places to optimize page loads

3.3.19 0.2.0 (Released 09-01-2011)

- First official release

Development sponsored by Caktus Consulting Group, LLC.

3.4 Contributing

django-timepiece is an open-source project and we are excited to have community contributions.

3.4.1 Submit an Issue

Issues are managed on Github. If you think you’ve found a bug then it’s helpful if you indicate the version of django-timepiece that you are using. If you think your bug is JavaScript- or styling-related, then it is also helpful to know which browser you are using.

Issues are also used to track new features. If you have a feature you would like to see then you can submit a proposal ticket. We love it when others add features and push them back up to us!

3.4.2 Get the Source

Feel free to fork django-timepiece and make your own changes. You can download the full source by cloning the git repo:

```
git clone https://github.com/caktus/django-timepiece.git
```

If you think that your changes could be helpful to others, please submit a pull request to have it merged in. Here’s a quick guide:

1. Fork the repo.
2. *Run the tests*. We only take pull requests with passing tests, and it’s great to know that you have a clean slate.

3. Add a test for your change. Only refactoring and documentation changes require no new tests. If you are adding functionality or fixing a bug, we need a test!
4. Make the test pass in all tox environments.
5. Follow PEP8 style conventions. Use 4 spaces instead of tabs.
6. We'll love you forever if you include documentation!
7. Push to your fork and submit a pull request.
8. Now you're waiting on us. We'll typically review and comment on your pull request within 3 business days. We may suggest some changes, improvements, or alternatives to be used before pulling in the changeset.

3.4.3 Contact Us

If you have any questions, feel free to reach out to us on GitHub or through our website.

CHAPTER

FOUR

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*