django-timepiece Documentation

Release 0.9

May 14, 2013

CONTENTS

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

master

develop

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 officially 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. If you are starting from the included example project, copy the example local settings file at *example_project/settings/local.py.example* to *example_project/settings/local.py*.

   If you are using an existing project, you will need to make the following changes to your settings:

   - Add *timepiece* and its dependencies to `INSTALLED_APPS`:

     ```
     INSTALLED_APPS = (
         ...
         ’bootstrap_toolkit’,
         ’compressor’,
         ’pagination’,
         ’selectable’,

         ’timepiece’,
         ’timepiece.reports’,
         ...
     )
     ```

   - 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.quick_clock_in",      # <----
     ```

```
        "timepiece.context_processors.quick_search",        # <----
    )
```

- Configure compressor settings:

  ```
  COMPRESS_PRECOMPILERS = (
      ('text/less', 'lessc {infile} {outfile}'),
  )
  COMPRESS_ROOT = '%s/static/' % PROJECT_PATH
  INTERNAL_IPS = ('127.0.0.1',)
  ```

- Set `USE_TZ` to `False`. django-timepiece does not currently support timezones.

4. Run `syncdb`.

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

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

6. 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'),
    ...
)
```

7. 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 Employee Time Tracking

django-timepiece allows employees to keep track of their time on projects.

3.1.1 Entries

Entries describe a period of work by a user on a particular project. Each user may only have one active entry at a time; no entries can overlap. Additionally, entries are limited to 12 hours in length.

user Foreign key to a User.

project Foreign key to a Project.

activity Foreign key to an Activity that describes the type of work that was done (e.g., development, estimation, planning)

location Foreign key to a Location that describes where the work was done (e.g., home, office)

status Foreign key to a ‘status’ Attribute (e.g., tracking whether entries have been verified for payroll purposes)

start_time When the entry started.

end_time When the entry ended. If the entry has no end_time, it is an active entry. Each user may only have one active entry at any point in time.

seconds_paused How long the entry was paused.

pause_time If `pause_time` is not `None`, then the entry is currently paused.

comments Optional description of this entry.

hours Total number of hours that this entry covers.

3.1.2 Clocking In and Out

Users can track their time through the Clock In and Clock Out views. If the user tries to clock in while another entry is active, the previous entry is clocked out at the second before the new entry is clocked in.

Users may pause their active entry by using the “Pause” button on the weekly dashboard. To resume the entry, the user must click the “Resume” button.

The user can clock into a specific project from anywhere on the site by selecting the project from the “Clock” pulldown on the navbar. This links to the Clock In view with that project and the most recent activity on that project already filled in on the form.

3.1.3 Weekly Dashboard View

The dashboard gives an overview of the logged-in user’s work this week.

Active Entry

Each user may be clocked in to only one entry at a time. The user can view the status of their current entry on the dashboard. From this section, the user can clock out, pause/resume, or edit the current entry, or clock in or switch to a new entry.

Overall Progress

The overall progress bar displays the number of hours that the user has worked this week (including the time so far on the active entry), out of the number of hours they are expected to work in a week (defined in the UserProfile model). If the user has gone over the expected number of hours, the overtime portion of the bar is displayed in red.

Project Progress

The project progress table shows how much the user has worked on each project, out of the hours they were assigned to work this week (given by the ProjectHours model). The name of the each project links to the Clock In view with that project and the most recent activity on that project already filled in on the form.

All Entries

The user can see a detailed view of all of their entries for the current week. This list includes all entries that end in the current week, and does not include the active entry. Each entry has links for editing and removal.

Online Users

The online users tab lists the active entries of all other users.

3.1.4 Monthly Ledger View

The ledger gives a summary of the user's work in a given month. At the end of the month, the user can verify their entries for payroll purposes.

3.2 Settings

All django-timepiece settings are optional. Default values are given in `timepiece.defaults` and can be overriden in your project's settings.

3.2.1 TIMEPIECE_DEFAULT_LOCATION_SLUG

Default `None`

This setting allows you to set an initial Location to associate with an entry in the Clock In form. The user can override the default choice by selecting another Location when clocking in.

If `TIMEPIECE_DEFAULT_LOCATION_SLUG` is not given, then then no initial value is used.

3.2.2 TIMEPIECE_PAID_LEAVE_PROJECTS

Default `{}`

This setting allows you to specify projects which people can clock in to that are not business-related. These projects will not be included in the total number of 'worked' hours. For example:

```
TIMEPIECE_PAID_LEAVE_PROJECTS = {
    'sick': 1,
    'vacation': 2,
}
```

where each key is an arbitrary slug for the project and each value is the primary key of the associated project.

3.2.3 TIMEPIECE_ACCCOUNTING_EMAILS

Default `[]`

When pending contract hours are created or changed, an email can be sent to notify someone. This setting is a list of the email addresses where those emails should be sent.

3.2.4 TIMEPIECE_EMAILS_USE_HTTPS

Default `True`

Whether links in emails that timepiece sends should use https://. The default is True, but if set to False, links will use http://.

3.3 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.

To run a subset of the Django tests for django-timepiece, you can pass their names to `run_tests.py` as you would for `django-admin.py test`, e.g. `run_tests.py timepiece.TestClassName [...]`.

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`.

3.4 Release Notes

3.4.1 0.9.0 (05-10-2013)

We have reorganized the django-timepiece code into 5 co-dependent apps to make the code more modular, readable, and updatable. To upgrade your installation while maintaining your existing data, please follow these guidelines:

1. Ensure that all existing migrations for django-timepiece are up to date.
2. Upgrade your django-timepiece installation.
3. Add `'timepiece'`, `'timepiece.contracts'`, `'timepiece.crm'`, `'timepiece.entries'`, and `'timepiece.reports'` to *INSTALLED_APPS* in your settings file.
4. Run the new migrations:

   ```
   ./manage.py migrate timepiece --delete-ghost-migrations
   ./manage.py migrate reports
   ./manage.py migrate contracts --fake
   ./manage.py migrate crm --fake
   ./manage.py migrate entries --fake
   ```

5. Remove all of your old *.pyc files, e.g. run something like *find . -name '*.pyc' -delete* in bash.
6. Remove stale ContentType and Permission objects. Note: Before doing this, take note of which timepiece permissions are in each of your auth Groups as these will need to be restored.

   ```
   # This also deletes associated timepiece permissions.
   ContentType.objects.filter(app_label='timepiece').delete()
   ```

7. Trigger the creation of new ContentType and Permission objects:

   ```
   from django.contrib.auth.management import create_permissions
   from django.contrib.contenttypes.management import update_contenttypes
   from django.db.models import get_app, get_models

   for app in ['timepiece', 'contracts', 'crm', 'entries', 'reports']:
       update_content_types(get_app(app), get_models())
       create_permissions(get_app(app), get_models(), 0)
   ```

8. Restore permissions to any auth Groups that you have created.

Related issues are in the 0.9.0 milestone.

- Reorganized app structure (see notes above)
- Removed existing migrations (see notes above)
- Dropped support for Django 1.3
- Added support for Django 1.5
- Removed *PROJECT_UNSET* from *ProjectContract.type* choices
- Use ellipsis after comment summary on dashboard so that comment doesn't appear cut off

3.4.2 0.8.3 (Released 03-27-2013)

Related issues are in the 0.8.3 milestone.

- Fixed invoice creation to separate out non-billable activities and not count them in the invoice totals.

- Added link to *Outstanding Hours* page from *Invoice Detail* page.
- Added schedule link to mobile navbar
- Added requirement that scheduled hours be > 0.
- Added tests for project hours.
- Fixed weekly schedule editor so project name changes stick.
- Clicking 'Clock Out' more than once gives 404 error.

3.4.3 0.8.2 (Released 01-25-2013)

Related issues are in the 0.8.2 milestone.

- Added permission requirements to view invoice list
- Added static files blocks to the base template

3.4.4 0.8.1 (Released 01-22-2013)

Related issues are in the 0.8.1 milestone.

- Restored *slug* field on RelationshipType

3.4.5 0.8.0 (Released 01-21-2013)

Related issues are in the 0.8.0 milestone.

Features

- Cleaned up the URL and template structure (This will break many existing bookmarks!)
- Removed the General Ledger report in favor of adding a summary by project on the Hourly Report page
- Default to showing entries from the previous week grouped by day on the Hourly Report
- Fall back to displaying username when a user's first & last name are unavailable
- Added name field to ProjectContract model
- Made ProjectContract <-> Project a many-to-many relationship
- Added additional information on ProjectContract detail page
- Added list of contracts on Project detail page
- Allow running a subset of tests through *runtests.py* (now in accordance with existing documentation)
- Created a *get_active_entry* utility which raises *ActiveEntryError* if a user has more than one active entry
- Permanent tabs for user time sheet tabs
- Upgrade less from 1.3.0 -> 1.3.3
- New model ContractHours allows tracking whether specific blocks of hours on a contract have been approved.

Bugfixes

- Prevent "None" from appearing under date headers on dashboard's All Entries tab
- Save Auth groups when adding/editing a user

- Include current GET parameters when using ‘next’ in a URL

Other Changes

- Removed unused methods from ProjectContract and ContractAssignment models
- Removed unused ContractMilestone model
- Removed unused AssignmentManager class
- Removed unused *slug* fields from Business & RelationshipType models
- Removed ProjectContract from Project admin
- Improved test coverage of template tags
- Changed references to person/people to user/users for consistency with data model
- Removed unused *clear_form.js*
- Used slightly darker highlight color for active project on dashboard’s Progress tab
- Removed paste styles from *styles.less*
- Updated contributing docs to indicate that pull requests should be made to *caktus:develop*
- Removed some unused images, renamed a couple of others.

3.4.6 0.7.3 (Released 01-07-2013)

Related issues are in the 0.7.3 milestone.

Features

- Row and column highlighting on weekly schedule
- Redirect regular users to schedule view from schedule edit (rather than redirecting to login)
- Use checkbox select multiple for editing groups on person add/edit forms
- Added “active” column to front-end user list & detail views
- Permanent links to dashboard tabs
- Dashboard project progress table
  - Highlight row of active project
  - Made width of bars relative to maximum worked or assigned hours
  - Show overtime bar for work on unassigned projects
- Dashboard “All Entries” tab
  - Moved “Add Entry” button to top right of page, and clock in dropdown
  - Split entries by day into separate tables, with a summary row
  - Added comment column, and included comment in row tooltip
  - Hide pause time unless it is greater than 0

Bugfixes

- Fixed bugs in handling filters on the hourly report
- Only summarize entries in the time period requested on hourly & billable reports (previously, entries for the entire week which includes the from date were included)

- Fixed bug which prevented projects being removed from the hourly report filter
- Keep GET parameters when deleting entry (allows proper redirection)
- Use `history.back()` on cancel buttons on clock in, clock out, and add entry pages
- Fixed floating point errors that caused project progress bars to display over two lines
- Prevent negative worked/assigned time on project progress bars
- Fix project progress bar behavior when worked = 0 and assigned = 0 (e.g., just after clocking into an unassigned project)
- Allow editing groups on person edit page
- Fixed subnav rendering on invoice pages

3.4.7 0.7.2 (Released 11-28-2012)

- Fixed test failures that resulted from changes to the display of project names when clocking time.

3.4.8 0.7.1 (Released 11-28-2012)

Related issues are in the 0.7.1 milestone.

- Fixed path to white Glyphicons
- Fixed duplicates in unverified list on Payroll Summary report
- Removed unused `timepiece/time-sheet/_entry_list.html` template
- Made `Business.name` field required
- Schema migration to add `Business.short_name` field
- Add `Business.get_display_name()` to retrieve first of `short_name` or `name`
- Show business short name with project name on the dashboard, clock in, clock out, and outstanding invoices pages
- Added `Entry.get_paused_seconds()` - gets total time paused on any entry, regardless of whether it is currently active or paused
- Removed `Entry.get_active_seconds()`
- Moved `Entry.get_seconds()` to `Entry.get_total_seconds()` - updated to get total worked seconds for any entry, regardless of whether it is currently active or paused, also taking into account the amount of time paused
- Dashboard tweaks and bug fixes
  - Fixed pause time bug
  - Fixed incorrect link name in mobile navbar
  - Fixed floating point errors in progress bar width calculations
  - Fixed overall progress bar styling when worked width = 0%
  - Fixed project progress bar responsiveness when resizing or zooming the page
  - Show overtime on project progress bars
  - Use dark green instead of red on overtime bars

  - Separated the "Project" and "Activity" columns in the all entries list
  - Include active entry in the all entries list
  - Increased the prominence of the active entry section
  - Show the current activity name in the active entry section
  - Removed link to the active project from the active entry section
  - Use "for" instead of "on" when describing entries

3.4.9 0.7.0 (Released 11-16-2012)

Features

- Added search to Project list view in admin
- Added project relationship information on Person detail view
- Updated the navigation bar
  - Added "Quick Clock In" pulldown to allow link to project-specific clock in form from anywhere on the site
  - Replaced "Dashboard" pulldown with a link to the user's monthly time sheet. The dashboard is accessible via the "Timepiece" link in the top left corner.
  - Renamed "Reports" dropdown to "Management", and moved link to the admin from the user pulldown
  - Moved "Online Users" info to weekly dashboard view & removed the `active_entries` context processor
  - Made search box smaller unless it is the focused element
  - Use user's first name instead of email address on user pulldown
- Redesigned the weekly dashboard view
  - Active entry section allows convenient summary & manipulation of the current entry
  - Visualization of overall progress (out of hours set in `UserProfile.hours_per_week`)
  - Visualization of hours worked on each project (out of ProjectHours assigned this week)
  - Use "humanized" hours display (1:30) rather than decimal (1.5)
- Added productivity report, which compares the hours worked on a project to the hours that were assigned to it

Bug Fixes

- Updated to latest version of Bootstrap
- Updated django-compressor from 1.1.2 -> 1.2 & updated run_tests settings to avoid masking primary errors in tests
- Set `USE_TZ = False` in example_project settings because we don't currently support use of timezones
- Added missing app and context processors to settings in example_project and run_tests
- Updated example_project settings & README to reflect that INTERNAL_IPS must be set in order to ensure that Bootstrap Glyphicons can be found
- Fixed bug when copying the previous week's ProjectHours entries to current week when entries for the current week already exist.
- Fixed bug when removing ProjectRelationship through the front end

Code Quality

- Renamed the 'timepiece-entries' URL to 'dashboard'
- Removed unnecessary settings from example_project and run_tests
- Split up settings files in example project to use base and local settings
- Removed unused jqplot library
- Moved `multiply` template tag to timepiece_tags and removed math_tags file
- Removed most of custom icon set in favor of Bootstrap's Glyphicons

3.4.10 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

3.4.11 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

3.4.12 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.

3.4.13 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.

3.4.14 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

3.4.15 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

3.4.16 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.

3.4.17 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

3.4.18 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.

3.4.19 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.

3.4.20 0.3.7 (Released 02-01-2012)

- Make create invoice page inclusive of date

3.4.21 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.

3.4.22 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.

3.4.23 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.

3.4.24 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

3.4.25 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.

3.4.26 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

3.4.27 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

3.4.28 0.2.0 (Released 09-01-2011)

- First official release

Development sponsored by Caktus Consulting Group, LLC.

3.5 Contributing

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

3.5.1 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!

3.5.2 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 to *caktus:develop*.
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.

3.5.3 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*