

---

# **Ulimi Time Tracker Documentation**

***Release 1.0***

**Susie, Chathan, Rahul, Elijah**

**Dec 12, 2018**



---

## Contents:

---

<b>1</b>	<b>Indices and tables</b>	<b>3</b>
	<b>Python Module Index</b>	<b>19</b>



This is a complete list of functions and classes, automatically generated by sphinx. You can search by index or modules.



- `genindex`
- `modindex`
- `search`

**class** `vms.views.ClientAdminInviteAcceptView` (*\*\*kwargs*)

Accept in invitation to become an admin for a client.

**form\_class**

alias of `vms.forms.ClientAdminInviteAcceptForm`

**form\_valid** (*form*)

Save the form and redirect the user to the detail page of the client that the created admin links to.

**Args:**

**form:** The validated form instance to save.

**Returns:** A redirect response sending the user to the detail view of the created admin's client.

**get\_form\_kwargs** ()

Get additional keyword arguments for the form.

**Returns:** The keyword arguments used to initialize the form.

**class** `vms.views.ClientCreateView` (*\*\*kwargs*)

Create a new client company.

**form\_class**

alias of `vms.forms.ClientCreateForm`

**form\_valid** (*form*)

Save the form and redirect the user to the newly created client's detail page.

**Returns:** A redirect response sending the user to the detail page of the newly created client.

**test\_func** ()

Ensure the requesting user is the administrator of a staffing agency.

**Returns:** A boolean indicating if the user should be allowed to access the view.

**class** vms.views.**ClientDetailView** (\*\*kwargs)

Retrieve information about a specific client.

**get\_context\_data** (\*\*kwargs)

Get additional context to render the template with.

**Args:**

**\*\*kwargs:** Additional values to be passed to the template.

**Returns:** A dictionary containing the context used to render the view's template.

**get\_object** (queryset=None)

Get the client whose slug is specified in the URL.

**Args:**

**queryset:** The queryset to choose a client from. If a queryset is not provided, all clients are searched.

**Returns:** The client with the slug given in the URL.

**class** vms.views.**ClientJobCreateView** (\*\*kwargs)

**form\_class**

alias of `vms.forms.ClientJobCreate`

**form\_valid** (form)

If the form is valid, redirect to the supplied URL.

**get\_context\_data** (\*\*kwargs)

Insert the form into the context dict.

**get\_form\_kwargs** ()

Return the keyword arguments for instantiating the form.

**class** vms.views.**ClientJobDetailView** (\*\*kwargs)

View or update the details of a specific client job.

**get\_object** (queryset=None)

Get the job specified by the client and job present in the URL.

**Args:**

**queryset:** An optional queryset to choose from. If no queryset is provided, all client jobs will be searched.

**Returns:** The client job specified by the client and jobs slugs in the URL.

**class** vms.views.**ClientJobListView**

List the jobs for a particular client.

**get\_context\_data** (\*, object\_list=None, \*\*kwargs)

Get the context to render the template with.

**Args:**

**object\_list:** The list of client jobs rendered by the view.

**\*\*kwargs:** Additional context variables for the template.

**Returns:** A dictionary containing the context used to render the view's template.

**get\_queryset** ()

Get the jobs for the specified client.



**Returns:** The jobs owned by the client whose slug is given in the URL.

```
class vms.views.ClockInView(**kwargs)
```

View for clocking in.

**form\_class**

alias of *vms.forms.ClockInForm*

**form\_valid**(*form*)

If the form is valid, redirect to the supplied URL.

**get\_form\_kwargs**()

Return the keyword arguments for instantiating the form.

```
class vms.views.ClockOutView(**kwargs)
```

View for clocking out.

**form\_class**

alias of *vms.forms.ClockOutForm*

**form\_valid**(*form*)

Save the valid form instance.

**Args:**

**form:** The valid form instance.

**Returns:** A redirect response for the user.

**get\_form\_kwargs**()

Add the employee as a parameter for the form.

**Returns:** The kwargs to instantiate the form instance with.

```
class vms.views.DashboardView(**kwargs)
```

```
class vms.views.EmployeeApplyView(**kwargs)
```

Apply a staffing agency employee to a client company.

**form\_class**

alias of *vms.forms.EmployeeApplyForm*

**form\_valid**(*form*)

Save the form and redirect back to the staffing agency detail view.

**Args:**

**form:** The form to save.

**Returns:** A response redirecting the user to the staffing agency's detail view.

**get\_context\_data**(\*\*kwargs)

Get additional context used to render the view's template.

**Args:** \*\*kwargs:

**Returns:** A dictionary containing the context used to render the view's template.

**get\_form\_kwargs**()

Get the keyword arguments to initialize the form with.

**Returns:** A dictionary containing the arguments to initialize the view's form instance with.

```
class vms.views.EmployeeApproveView(**kwargs)
```

Can link a supervisor to an approved employee.

```

form_class
    alias of vms.forms.EmployeeApprovalForm

form_valid (form)
    If the form is valid, redirect to the supplied URL.

get_form_kwargs ()
    Return the keyword arguments for instantiating the form.

```

**class** `vms.views.EmployeeDetailView` (*\*\*kwargs*)  
View the details of a single employee.

```

get_context_data (**kwargs)
    Add date context to the view.

Args:

    **kwargs: Keyword arguments to pass to the base method.

Returns: A dictionary containing context used to render the view.

get_object (queryset=None)
    Get the employee using the parameters in the URL.

Args:

    queryset: The queryset to pull from. If not provided, all employees are searched.

Returns: The employee with the ID and client slug specified in the URL.

```

**class** `vms.views.PendingEmployeesView` (*\*\*kwargs*)  
Can see the list of pending employees that need to be approved or stopped

```

get_queryset ()
    Get the jobs for the specified client.

Returns: The jobs owned by the client whose slug is given in the URL.

```

**class** `vms.views.StaffingAgencyDetailView` (*\*\*kwargs*)  
Retrieve information about a specific staffing agency.

```

get_context_data (**kwargs)
    Insert the single object into the context dict.

get_object (**kwargs)
    Return the object the view is displaying.

    Require self.queryset and a pk or slug argument in the URLconf. Subclasses can override this to return any object.

```

**class** `vms.views.StaffingAgencyEmployeeApproveView` (*\*\*kwargs*)  
Approve an employee who has applied to a staffing agency.

```

form_class
    alias of vms.forms.StaffingAgencyEmployeeApprovalForm

form_valid (form)
    Save the form and redirect the user.

Args:

    form: The form to save.

Returns: A response redirecting the user. If a next parameter is provided in the URL, the user is redirected there. Otherwise they are redirected to the list of pending employees.

```

**get\_form\_kwargs()**

**Returns:** A dictionary containing the keyword arguments used to instantiate the view's form class.

**class** vms.views.StaffingAgencyEmployeeCreateView(\*\*kwargs)

Apply as an employee to a staffing agency.

**form\_class**

alias of *vms.forms.StaffingAgencyEmployeeCreateForm*

**form\_valid(form)**

Save the form and redirect the user.

**Args:**

**form:** The form to save.

**Returns:** A response redirecting the user to their dashboard.

**get\_form\_kwargs()**

Get the keyword arguments used to instantiate the form.

**Returns:** A dictionary containing the arguments used to instantiate an instance of the view's form class.

**class** vms.views.StaffingAgencyEmployeeDetailView(\*\*kwargs)

**get\_context\_data(\*\*kwargs)**

Get context data to render the view's template with.

**Args:** \*\*kwargs:

**Returns:** A dictionary containing context to render the view's template with.

**get\_object(\*args, \*\*kwargs)**

**Returns:** The staffing agency employee whose ID is specified in the URL.

**class** vms.views.StaffingAgencyEmployeePendingListView(\*args, \*\*kwargs)

List the pending employees for a staffing agency.

**get\_context\_data(\*args, \*\*kwargs)**

**Returns:** The context used to render the view's template.

**get\_queryset()**

**Returns:** A queryset containing the pending employees for the staffing agency whose slug is given in the URL.

**class** vms.views.TimeRecordApproveView(\*\*kwargs)

Approve a specific time record.

**form\_class**

alias of *vms.forms.TimeRecordApprovalForm*

**form\_valid(form)**

Save the form and redirect back to the hours approval page.

**Args:**

**form:** The valid form instance to save.

**Returns:** A redirect response. If a 'next' parameter is provided in the URL, the user is taken to that URL. Otherwise they are taken to the list of unapproved hours for the client who owns the time record that was just approved.

**get\_form\_kwargs()**

Add additional information required for our custom form.

**Returns:** The keyword arguments used to construct the form.

**class vms.views.UnapprovedTimeRecordListView(\*\*kwargs)**

List all the unapproved time records for a specific client.

**get\_queryset()**

Get the list of unapproved hours for the client.

**Returns:** A queryset containing the unapproved time records for the client specified in the URL.

**class vms.models.Client(\*args, \*\*kwargs)**

A client is a company that employees perform work for.

**exception DoesNotExist**

**exception MultipleObjectsReturned**

**clean()**

Generate a unique ID and slug if necessary.

**get\_absolute\_url()**

Get the URL of the instance's detail view.

**Returns:** The absolute URL of the instance's detail view.

**job\_list\_url**

Get the URL of the client's job list.

**Returns:** The URL of the view where all the client's jobs are listed.

**unapproved\_time\_record\_list\_url**

Get the URL of the client's unapproved time record list.

**Returns:** The absolute URL of the view to list the client's unapproved time records.

**class vms.models.ClientAdmin(\*args, \*\*kwargs)**

A link between a user and a client that grants the linked user admin permissions on the associated client company.

**exception DoesNotExist**

**exception MultipleObjectsReturned**

**class vms.models.ClientAdminInvite(\*args, \*\*kwargs)**

An invitation for a user to become a client admin.

**exception DoesNotExist**

**exception MultipleObjectsReturned**

**accept(user)**

Accept the invitation.

**Args:**

**user:** The user who accepted the invitation.

**Returns:** The created client admin.

**accept\_url**

**Returns:** The absolute URL of the view to accept the invite.

**send** (*request*)

Send an invitation message to the email address associated with the instance.

**Args:**

**request:** The request that was made to trigger the send. This is used to build the full URL for accepting the invite.

**class** vms.models.**ClientJob** (*\*args, \*\*kwargs*)

A type of task that can be worked on for a specific client.

**exception** **DoesNotExist**

**exception** **MultipleObjectsReturned**

**clean** ()

Generate a slug for the instance if necessary.

**clean\_fields** (*exclude=None*)

Ensure the name of the instance is not too similar to other jobs owned by the same client.

**Args:**

**exclude:** An optional list of fields to exclude from validation. Defaults to `None`.

**get\_absolute\_url** ()

Get the absolute URL of the instance's detail view.

**Returns:** The absolute URL of the instance's detail view.

**class** vms.models.**Employee** (*\*args, \*\*kwargs*)

An employee working for a specific company.

**exception** **DoesNotExist**

**exception** **MultipleObjectsReturned**

**approve** (*admin*)

Approve the employee's request to join the client.

**Args:**

**admin:** The client admin who approved the request.

**approve\_url**

**Returns:** The absolute URL of the view used to approve the employee.

**clock\_in\_url**

**Returns:** The absolute URL of the view used to clock in the employee.

**clock\_out\_url**

**Returns:** The absolute URL of the view used to clock out the employee.

**get\_absolute\_url** ()

**Returns:** The URL of the view for an employee of a client

**is\_clocked\_in**

Determine if the employee is currently clocked in.

**Returns:** A boolean indicating if the employee is clocked in.

**save** (*\*args, \*\*kwargs*)

Save the employee and generate an ID for them if necessary.

**total\_time**

Get the total time that the employee has logged.

**Returns:** The total time the employee has worked, in seconds.

**validate\_unique** (*exclude=None*)

Validate that the employee's ID is unique to the company they work for.

**Args:**

**exclude:** An optional list of field names to exclude from the check.

**class** vms.models.StaffingAgency (\*args, \*\*kwargs)

A company that provides employees to clients.

**exception** DoesNotExist

**exception** MultipleObjectsReturned

**get\_absolute\_url** ()

**Returns:** The absolute URL of the instance's detail view.

**save** (\*args, \*\*kwargs)

Save the agency, creating a slug if necessary.

**Args:**

**\*args:** Positional arguments to pass to the original save method.

**\*\*kwargs:** Keyword arguments to pass to the original save method.

**class** vms.models.StaffingAgencyAdmin (\*args, \*\*kwargs)

A link between a user and a staffing agency that grants the linked user admin permissions on the associated staffing agency.

**exception** DoesNotExist

**exception** MultipleObjectsReturned

**class** vms.models.StaffingAgencyEmployee (\*args, \*\*kwargs)

An employee who is contracted out by a staffing agency.

**exception** DoesNotExist

**exception** MultipleObjectsReturned

**approve** (*admin*)

Mark the staffing agency employee as approved.

**Args:**

**admin:** The staffing agency administrator who approved the employee.

**get\_absolute\_url** ()

Get the URL of the staffing agency's employee.

**Returns:** The URL of the view where the staffing agency's employee information is listed.

**class** vms.models.TimeRecord (\*args, \*\*kwargs)

A record marking a work period for an employee.

**exception** DoesNotExist

**exception** MultipleObjectsReturned

**approval\_url**

Get the URL of the view used to approve the time record.

**Returns:** The absolute URL of the view used to approve the time record.

**is\_approved**

Determine if the time record is approved.

**Returns:** A boolean indicating if there is an approval record for the time record.

**projected\_earnings**

Get the projected earning from this time record.

**Returns:** The total time multiplied by the pay rate for this job.

**total\_time**

Get the total time from this time record. Returns:

The time delta between start time and end time or delta of 0 if no end time.

**class** vms.models.TimeRecordApproval(\*args, \*\*kwargs)

A record of the approval for a time record.

**exception** DoesNotExist

**exception** MultipleObjectsReturned

vms.models.generate\_slug(value, queryset, slug\_dest='slug')

Generate and save a unique slug for the provided instance.

**Args:**

**value:** The value to slugify.

**queryset:** The queryset to search to ensure the generated slug is unique.

**slug\_dest:** The name of the attribute on the instance that the slug is saved to.

vms.models.generate\_token()

Generate a random token.

**Returns:** A random 16-character alphanumeric string.

**class** vms.mixins.DateRangeMixin

Mixin providing functionality for filtering by a date range.

The starting and ending times are specified as GET parameters.

**end\_date**

**Returns:** The end date provided in the URL. If both a start and end date are specified and the end date is prior to the start date, the start date is returned instead.

Note the end date is specified as a datetime so that we can include results for the entire day. This results in an inclusive bound which is more intuitive.

**filter\_by\_date**(queryset, start\_attr='time\_start', end\_attr='time\_end')

Filter a queryset based on a date range.

If either bound is not provided, it is not restricted.

**Args:**

**queryset:** The queryset to filter.

**start\_attr:** The attribute on the queryset that should be compared to the start time given in the URL.

**end\_attr:** The attribute on the queryset that should be compared to the end time given in the URL.

**Returns:** The provided queryset filtered such that no record's `start_attr` is before the start time in the URL and no record's `end_attr` is after the end time in the URL.

**get\_context\_data** (*\*\*kwargs*)

Add date context to the view.

**Args:**

**\*\*kwargs:** Keyword arguments to pass to the base method.

**Returns:** A dictionary containing context used to render the view.

**start\_date**

**Returns:** The start date provided in the URL as a datetime instance.

`vms.time_utils.round_time_worked` (*time\_worked*, *block\_size=900*)

Round a period of time worked to the nearest block.

Typically this is used to round time worked to the closest 15 minute block.

**Args:**

**time\_worked:** The time period to round, in seconds.

**block\_size:** The block size to round to, in seconds. Defaults to 15 minutes.

**Returns:** The time worked, in seconds, rounded to the nearest block.

**class** `vms.managers.TimeRecordQuerySet` (*model=None*, *query=None*, *using=None*,  
*hints=None*)

**total\_time** ()

Get the total duration of the time records in the queryset.

**Returns:** The total duration of the time records in the queryset expressed as a `datetime.timedelta` instance.

**with\_deltas** ()

Annotate the queryset to include a delta for each time record.

Because computing a delta relies on `time_end` being present, time records that have not been completed are excluded.

**Returns:** A queryset annotated such that each time record has a `delta` attribute containing the delta between the record's `time_start` and `time_end`.

`vms.id_utils.generate_numeric_id` (*num\_digits*)

Generate a numeric ID with the provided number of digits.

**Args:**

**num\_digits:** The number of digits present in the returned ID.

**Returns:** A random ID with the specified number of digits.

`vms.id_utils.generate_unique_id` (*digits*, *queryset*, *queryset\_attr='id'*)

Generate a unique ID for a given queryset.

The provided ID is guaranteed to be unique for the provided queryset.

warning ..

There is still a race condition between when the value is returned from the function and when it is saved to an instance.

**Args:**

**digits:** The number of digits in the returned ID.



**queryset:** The queryset used to check for uniqueness.

**queryset\_attr:** The attribute of the queryset to check for uniqueness. Defaults to `id`.

**Returns:** A unique ID for the provided queryset of objects.

**class** `vms.forms.ClientAdminInviteAcceptForm` (*token, \*args, \*\*kwargs*)

Form to accept an invitation to become a client admin.

**clean()**

Ensure the provided token is valid.

**save** (*user*)

Accept the invitation with the provided token.

**Args:**

**user:** The user who is accepting the invitation.

**Returns:** The newly created `ClientAdmin` instance.

**class** `vms.forms.ClientCreateForm` (*data=None, files=None, auto\_id='id\_%s', pre-  
fix=None, initial=None, error\_class=<class  
'django.forms.utils.ErrorList'>, label\_suffix=None,  
empty\_permitted=False, instance=None,  
use\_required\_attribute=None, renderer=None*)

Form to create a new client.

Once the form is saved, a new client company is saved, and an email invitation to become an admin for the client is sent to the provided email address.

**save** (*request=None, commit=True*)

Create a new client and send an email to the admin.

**Args:**

**request:** The request made to trigger the form save.

**commit:** A boolean indicating if the created model should be saved to the database or not.

**Returns:** The created client instance.

**class** `vms.forms.ClientJobCreate` (*client, \*args, \*\*kwargs*)

Form to create a new client job.

**class** `vms.forms.ClockInForm` (*employee, \*args, \*\*kwargs*)

Form to clock in the requesting user.

**clean()**

Validate the form to ensure the employee is not already clocked in.

**save()**

Save the form to create a new time record.

**class** `vms.forms.ClockOutForm` (*employee, \*args, \*\*kwargs*)

Form to clock out an employee.

**clean()**

Validate the form's data.

**Raises:**

**forms.ValidationError:** If the employee is not clocked in.

**save()**

Complete the employee's open time record.

**class** vms.forms.**EmployeeApplyForm** (*staffing\_agency, user, \*args, \*\*kwargs*)

Form for applying an employee to a client company.

**save** (*commit=True*)

Save the new employee and attach the fields provided when the form was created.

**Args:**

**commit:** A boolean indicating if the created employee should be saved to the database. Defaults to True.

**Returns:** The created employee.

**class** vms.forms.**EmployeeApprovalForm** (*employee, \*args, \*\*kwargs*)

Form to link Employee to Supervisor

**class** vms.forms.**StaffingAgencyEmployeeApprovalForm** (*employee, admin, \*args, \*\*kwargs*)

Form to approve a staff employee

**clean** ()

Ensure the employee and admin are associated with the same staffing agency.

**Returns:** The cleaned data.

**save** ()

Approve the staffing agency employee associated with the form.

**class** vms.forms.**StaffingAgencyEmployeeCreateForm** (*user, \*args, \*\*kwargs*)

Form to create a staffing agency employee.

**save** (*user*)

Create a new staffing agency employee.

**Args:**

**user:** The user to associate with the employee that is created.

**Returns:** The newly created staffing agency employee.

**class** vms.forms.**TimeRecordApprovalForm** (*time\_record, approving\_user, \*args, \*\*kwargs*)

Form to approve a time record.

**save** ()

Create an approval for the time record associated with the form.

**Returns:** The approval instance created for the time record.

**class** vms.admin.**ClientAdmin** (*model, admin\_site*)

**class** vms.admin.**ClientAdminAdmin** (*model, admin\_site*)

**class** vms.admin.**ClientAdminInviteAdmin** (*model, admin\_site*)

**class** vms.admin.**ClientJobAdmin** (*model, admin\_site*)

**class** vms.admin.**EmployeeAdmin** (*model, admin\_site*)

**supervisor\_name** (*obj*)

Get the name of the employee's supervisor.

**Args:**

**obj:** The employee to get the supervisor of.

**Returns:** The name of the employee's supervisor if they have an assigned supervisor. Otherwise, a placeholder empty value is returned.

**class** vms.admin.StaffingAgencyAdmin(*model, admin\_site*)

**class** vms.admin.StaffingAgencyAdminAdmin(*model, admin\_site*)

**class** vms.admin.StaffingAgencyEmployeeAdmin(*model, admin\_site*)

**class** vms.admin.TimeRecordAdmin(*model, admin\_site*)

**class** vms.admin.TimeRecordApprovalAdmin(*model, admin\_site*)

vms.api.dialogflow.**clock\_in**(*client\_id, employee\_id, project\_id=None*)  
Clock in an employee.

**Args:**

**client\_id:** The ID of the client the employee works for.

**employee\_id:** The ID of the employee who is clocking in.

**project\_id:** The ID of the project the employee is working on.

**Returns:** A response indicating the result of the clock in request.

vms.api.dialogflow.**clock\_out**(*client\_id, employee\_id*)  
Clock out the employee with the provided information.

**Args:**

**client\_id:** The ID of the client that the employee who is clocking out works for.

**employee\_id:** The ID of the employee to clock out.

**Returns:** A string describing the result of the clock out request.

vms.api.dialogflow.**list\_projects**(*client\_id*)  
Enumerate the projects for a client.

**Args:**

**client\_id:** The ID of the client company whose projects should be listed.

**Returns:** A string message enumerating the projects.

vms.api.dialogflow.**process**(*data*)  
Fulfill a webhook request from Dialogflow.

**Args:**

**data:** The data received from Dialogflow.

**Returns:** A dictionary containing the data to return to Dialogflow.

**class** vms.api.serializers.DialogflowWebhookSerializer(*instance=None, data=<class 'rest\_framework.fields.empty'>, \*\*kwargs*)

Serializer to handle a request from Dialogflow.

**class** vms.api.serializers.IntentSerializer(*instance=None, data=<class 'rest\_framework.fields.empty'>, \*\*kwargs*)  
Serializer to handle information about the intent triggered by Dialogflow.

**class** vms.api.serializers.QuerySerializer(*instance=None, data=<class 'rest\_framework.fields.empty'>, \*\*kwargs*)  
Serializer to handle the query parameters for a Dialogflow fulfillment request.

```
class vms.api.views.DialogflowFulfillmentView(**kwargs)

    serializer_class
        alias of vms.api.serializers.DialogflowWebhookSerializer

class account.admin.UserAdmin(model, admin_site)

class account.forms.SignUpForm(*args, **kwargs)

class account.managers.UserManager
    Manager object for custom user class.

    create_superuser(*args, **kwargs)
        Create a user with superuser privileges.

    Args:

        *args: The positional args to pass to create_user.

        **kwargs: The keyword args to pass to create_user.

    Returns: The new user instance.

    create_user(name, username, password=None, **kwargs)
        Create a new user.

    Args:

        name: The user's full name.

        username: The user's username.

        password: The user's password.

        **kwargs: Additional keyword arguments to pass to the model instance.

    Returns: The new user instance.

class account.middleware.TimezoneMiddleware(get_response)
    Class to set the timezone for each request.

class account.models.User(*args, **kwargs)
    A single user.

    exception DoesNotExist

    exception MultipleObjectsReturned

class account.views.ProfileView(**kwargs)
    View the authenticated user's profile.

class account.views.SignUpView(**kwargs)
    View to register new users.

    form_class
        alias of account.forms.SignUpForm

    form_valid(form)
        Save the signup form and log in the created user.

    Args:

        form: The form containing the data sent to the view.

    Returns: A redirect response sending the new user to their profile page.
```

**get\_success\_url()**

Get the URL that the user is redirected to after submitting the form.

**Returns:** The URL provided in the 'next' parameter or the URL of the user's profile if no redirect URL is provided.



### a

`account.admin`, 16  
`account.forms`, 16  
`account.managers`, 16  
`account.middleware`, 16  
`account.models`, 16  
`account.views`, 16

### v

`vms.admin`, 14  
`vms.api.dialogflow`, 15  
`vms.api.serializers`, 15  
`vms.api.views`, 15  
`vms.forms`, 13  
`vms.id_utils`, 12  
`vms.managers`, 12  
`vms.mixins`, 11  
`vms.models`, 8  
`vms.time_utils`, 12  
`vms.views`, 3





## A

accept() (vms.models.ClientAdminInvite method), 8  
 accept\_url (vms.models.ClientAdminInvite attribute), 8  
 account.admin (module), 16  
 account.forms (module), 16  
 account.managers (module), 16  
 account.middleware (module), 16  
 account.models (module), 16  
 account.views (module), 16  
 approval\_url (vms.models.TimeRecord attribute), 10  
 approve() (vms.models.Employee method), 9  
 approve() (vms.models.StaffingAgencyEmployee method), 10  
 approve\_url (vms.models.Employee attribute), 9

## C

clean() (vms.forms.ClientAdminInviteAcceptForm method), 13  
 clean() (vms.forms.ClockInForm method), 13  
 clean() (vms.forms.ClockOutForm method), 13  
 clean() (vms.forms.StaffingAgencyEmployeeApprovalForm method), 14  
 clean() (vms.models.Client method), 8  
 clean() (vms.models.ClientJob method), 9  
 clean\_fields() (vms.models.ClientJob method), 9  
 Client (class in vms.models), 8  
 Client.DoesNotExist, 8  
 Client.MultipleObjectsReturned, 8  
 ClientAdmin (class in vms.admin), 14  
 ClientAdmin (class in vms.models), 8  
 ClientAdmin.DoesNotExist, 8  
 ClientAdmin.MultipleObjectsReturned, 8  
 ClientAdminAdmin (class in vms.admin), 14  
 ClientAdminInvite (class in vms.models), 8  
 ClientAdminInvite.DoesNotExist, 8  
 ClientAdminInvite.MultipleObjectsReturned, 8  
 ClientAdminInviteAcceptForm (class in vms.forms), 13  
 ClientAdminInviteAcceptView (class in vms.views), 3  
 ClientAdminInviteAdmin (class in vms.admin), 14

ClientCreateForm (class in vms.forms), 13  
 ClientCreateView (class in vms.views), 3  
 ClientDetailView (class in vms.views), 4  
 ClientJob (class in vms.models), 9  
 ClientJob.DoesNotExist, 9  
 ClientJob.MultipleObjectsReturned, 9  
 ClientJobAdmin (class in vms.admin), 14  
 ClientJobCreate (class in vms.forms), 13  
 ClientJobCreateView (class in vms.views), 4  
 ClientJobDetailView (class in vms.views), 4  
 ClientJobListView (class in vms.views), 4  
 clock\_in() (in module vms.api.dialogflow), 15  
 clock\_in\_url (vms.models.Employee attribute), 9  
 clock\_out() (in module vms.api.dialogflow), 15  
 clock\_out\_url (vms.models.Employee attribute), 9  
 ClockInForm (class in vms.forms), 13  
 ClockInView (class in vms.views), 5  
 ClockOutForm (class in vms.forms), 13  
 ClockOutView (class in vms.views), 5  
 create\_superuser() (account.managers.UserManager method), 16  
 create\_user() (account.managers.UserManager method), 16

## D

DashboardView (class in vms.views), 5  
 DateRangeMixin (class in vms.mixins), 11  
 DialogflowFulfillmentView (class in vms.api.views), 15  
 DialogflowWebhookSerializer (class in vms.api.serializers), 15

## E

Employee (class in vms.models), 9  
 Employee.DoesNotExist, 9  
 Employee.MultipleObjectsReturned, 9  
 EmployeeAdmin (class in vms.admin), 14  
 EmployeeApplyForm (class in vms.forms), 14  
 EmployeeApplyView (class in vms.views), 5  
 EmployeeApprovalForm (class in vms.forms), 14  
 EmployeeApproveView (class in vms.views), 5

EmployeeDetailView (class in vms.views), 6  
end\_date (vms.mixins.DateRangeMixin attribute), 11

## F

filter\_by\_date() (vms.mixins.DateRangeMixin method), 11  
form\_class (account.views.SignUpView attribute), 16  
form\_class (vms.views.ClientAdminInviteAcceptView attribute), 3  
form\_class (vms.views.ClientCreateView attribute), 3  
form\_class (vms.views.ClientJobCreateView attribute), 4  
form\_class (vms.views.ClockInView attribute), 5  
form\_class (vms.views.ClockOutView attribute), 5  
form\_class (vms.views.EmployeeApplyView attribute), 5  
form\_class (vms.views.EmployeeApproveView attribute), 5  
form\_class (vms.views.StaffingAgencyEmployeeApproveView attribute), 6  
form\_class (vms.views.StaffingAgencyEmployeeCreateView attribute), 7  
form\_class (vms.views.TimeRecordApproveView attribute), 7  
form\_valid() (account.views.SignUpView method), 16  
form\_valid() (vms.views.ClientAdminInviteAcceptView method), 3  
form\_valid() (vms.views.ClientCreateView method), 3  
form\_valid() (vms.views.ClientJobCreateView method), 4  
form\_valid() (vms.views.ClockInView method), 5  
form\_valid() (vms.views.ClockOutView method), 5  
form\_valid() (vms.views.EmployeeApplyView method), 5  
form\_valid() (vms.views.EmployeeApproveView method), 6  
form\_valid() (vms.views.StaffingAgencyEmployeeApproveView method), 6  
form\_valid() (vms.views.StaffingAgencyEmployeeCreateView method), 7  
form\_valid() (vms.views.TimeRecordApproveView method), 7

## G

generate\_numeric\_id() (in module vms.id\_utils), 12  
generate\_slug() (in module vms.models), 11  
generate\_token() (in module vms.models), 11  
generate\_unique\_id() (in module vms.id\_utils), 12  
get\_absolute\_url() (vms.models.Client method), 8  
get\_absolute\_url() (vms.models.ClientJob method), 9  
get\_absolute\_url() (vms.models.Employee method), 9  
get\_absolute\_url() (vms.models.StaffingAgency method), 10  
get\_absolute\_url() (vms.models.StaffingAgencyEmployee method), 10

get\_context\_data() (vms.mixins.DateRangeMixin method), 11  
get\_context\_data() (vms.views.ClientDetailView method), 4  
get\_context\_data() (vms.views.ClientJobCreateView method), 4  
get\_context\_data() (vms.views.ClientJobListView method), 4  
get\_context\_data() (vms.views.EmployeeApplyView method), 5  
get\_context\_data() (vms.views.EmployeeDetailView method), 6  
get\_context\_data() (vms.views.StaffingAgencyDetailView method), 6  
get\_context\_data() (vms.views.StaffingAgencyEmployeeDetailView method), 7  
get\_context\_data() (vms.views.StaffingAgencyEmployeePendingListView method), 7  
get\_form\_kwargs() (vms.views.ClientAdminInviteAcceptView method), 3  
get\_form\_kwargs() (vms.views.ClientJobCreateView method), 4  
get\_form\_kwargs() (vms.views.ClockInView method), 5  
get\_form\_kwargs() (vms.views.ClockOutView method), 5  
get\_form\_kwargs() (vms.views.EmployeeApplyView method), 5  
get\_form\_kwargs() (vms.views.EmployeeApproveView method), 6  
get\_form\_kwargs() (vms.views.StaffingAgencyEmployeeApproveView method), 6  
get\_form\_kwargs() (vms.views.StaffingAgencyEmployeeCreateView method), 7  
get\_form\_kwargs() (vms.views.TimeRecordApproveView method), 7  
get\_object() (vms.views.ClientDetailView method), 4  
get\_object() (vms.views.ClientJobDetailView method), 4  
get\_object() (vms.views.EmployeeDetailView method), 6  
get\_object() (vms.views.StaffingAgencyDetailView method), 6  
get\_object() (vms.views.StaffingAgencyEmployeeDetailView method), 7  
get\_queryset() (vms.views.ClientJobListView method), 4  
get\_queryset() (vms.views.PendingEmployeesView method), 6  
get\_queryset() (vms.views.StaffingAgencyEmployeePendingListView method), 7  
get\_queryset() (vms.views.UnapprovedTimeRecordListView method), 8  
get\_success\_url() (account.views.SignUpView method), 16

IntentSerializer (class in vms.api.serializers), 15

is\_approved (vms.models.TimeRecord attribute), 11  
is\_clocked\_in (vms.models.Employee attribute), 9

## J

job\_list\_url (vms.models.Client attribute), 8

## L

list\_projects() (in module vms.api.dialogflow), 15

## P

PendingEmployeesView (class in vms.views), 6  
process() (in module vms.api.dialogflow), 15  
ProfileView (class in account.views), 16  
projected\_earnings (vms.models.TimeRecord attribute), 11

## Q

QuerySerializer (class in vms.api.serializers), 15

## R

round\_time\_worked() (in module vms.time\_utils), 12

## S

save() (vms.forms.ClientAdminInviteAcceptForm method), 13  
save() (vms.forms.ClientCreateForm method), 13  
save() (vms.forms.ClockInForm method), 13  
save() (vms.forms.ClockOutForm method), 13  
save() (vms.forms.EmployeeApplyForm method), 14  
save() (vms.forms.StaffingAgencyEmployeeApprovalForm method), 14  
save() (vms.forms.StaffingAgencyEmployeeCreateForm method), 14  
save() (vms.forms.TimeRecordApprovalForm method), 14  
save() (vms.models.Employee method), 9  
save() (vms.models.StaffingAgency method), 10  
send() (vms.models.ClientAdminInvite method), 8  
serializer\_class (vms.api.views.DialogflowFulfillmentView attribute), 16  
SignUpForm (class in account.forms), 16  
SignUpView (class in account.views), 16  
StaffingAgency (class in vms.models), 10  
StaffingAgency.DoesNotExist, 10  
StaffingAgency.MultipleObjectsReturned, 10  
StaffingAgencyAdmin (class in vms.admin), 15  
StaffingAgencyAdmin (class in vms.models), 10  
StaffingAgencyAdmin.DoesNotExist, 10  
StaffingAgencyAdmin.MultipleObjectsReturned, 10  
StaffingAgencyAdminAdmin (class in vms.admin), 15  
StaffingAgencyDetailView (class in vms.views), 6  
StaffingAgencyEmployee (class in vms.models), 10  
StaffingAgencyEmployee.DoesNotExist, 10

StaffingAgencyEmployee.MultipleObjectsReturned, 10  
StaffingAgencyEmployeeAdmin (class in vms.admin), 15  
StaffingAgencyEmployeeApprovalForm (class in vms.forms), 14  
StaffingAgencyEmployeeApproveView (class in vms.views), 6  
StaffingAgencyEmployeeCreateForm (class in vms.forms), 14  
StaffingAgencyEmployeeCreateView (class in vms.views), 7  
StaffingAgencyEmployeeDetailView (class in vms.views), 7  
StaffingAgencyEmployeePendingListView (class in vms.views), 7  
start\_date (vms.mixins.DateRangeMixin attribute), 12  
supervisor\_name() (vms.admin.EmployeeAdmin method), 14

## T

test\_func() (vms.views.ClientCreateView method), 3  
TimeRecord (class in vms.models), 10  
TimeRecord.DoesNotExist, 10  
TimeRecord.MultipleObjectsReturned, 10  
TimeRecordAdmin (class in vms.admin), 15  
TimeRecordApproval (class in vms.models), 11  
TimeRecordApproval.DoesNotExist, 11  
TimeRecordApproval.MultipleObjectsReturned, 11  
TimeRecordApprovalAdmin (class in vms.admin), 15  
TimeRecordApprovalForm (class in vms.forms), 14  
TimeRecordApproveView (class in vms.views), 7  
TimeRecordQuerySet (class in vms.managers), 12  
TimezoneMiddleware (class in account.middleware), 16  
total\_time (vms.models.Employee attribute), 9  
total\_time (vms.models.TimeRecord attribute), 11  
total\_time() (vms.managers.TimeRecordQuerySet method), 12

## U

unapproved\_time\_record\_list\_url (vms.models.Client attribute), 8  
UnapprovedTimeRecordListView (class in vms.views), 8  
User (class in account.models), 16  
User.DoesNotExist, 16  
User.MultipleObjectsReturned, 16  
UserAdmin (class in account.admin), 16  
UserManager (class in account.managers), 16

## V

validate\_unique() (vms.models.Employee method), 10  
vms.admin (module), 14  
vms.api.dialogflow (module), 15  
vms.api.serializers (module), 15  
vms.api.views (module), 15  
vms.forms (module), 13

[vms.id\\_utils \(module\)](#), [12](#)  
[vms.managers \(module\)](#), [12](#)  
[vms.mixins \(module\)](#), [11](#)  
[vms.models \(module\)](#), [8](#)  
[vms.time\\_utils \(module\)](#), [12](#)  
[vms.views \(module\)](#), [3](#)

## W

[with\\_deltas\(\)](#) ([vms.managers.TimeRecordQuerySet](#)  
method), [12](#)