☰   ☁   🔍 👤

Developers   ⌄

Ideas Class   ☰

Ideas Class

Represents zone ideas.

Namespace

System

Usage

Ideas is a zone of users who post, vote for, and comment on ideas. An Ideas zone provides an online, transparent way for you to attract, manage, and showcase innovation.

A set of *recent replies* (returned by methods, see below) includes ideas that a user posted or commented on that already have comments posted by another user. The returned ideas are listed based on the time of the last comment made by another user, with the most recent ideas appearing first.

The *userID* argument is a required argument that filters the results so only the ideas that the specified user has posted or commented on are returned.

The *communityID* argument filters the results so only the ideas within the specified zone are returned. If this argument is the empty string, then all recent replies for the specified user are returned regardless of the zone.

For more information on ideas, see "Using Ideas" in the Salesforce online help.

Example

The following example finds ideas in a specific zone that have similar titles as a new idea:

```
public class FindSimilarIdeasController {

    public static void test() {
        // Instantiate a new idea
        Idea idea = new Idea ();

        // Specify a title for the new idea
        idea.Title = 'Increase Vacation Time for Employees';

        // Specify the communityID (INTERNAL_IDEAS) in which to find similar ideas.
        Community community = [ SELECT Id FROM Community WHERE Name = 'INTERNAL_IDEAS' ];

        idea.CommunityId = community.Id;

        ID[] results = Ideas.findSimilar(idea);
    }
}
```

The following example uses a Visualforce page in conjunction with a *custom controller*, that is, a special Apex class. For more information on Visualforce, see the Visualforce Developer's Guide.

This example creates an Apex method in the controller that returns unread recent replies. You can leverage this same example for the `getAllRecentReplies` and `getReadRecentReplies` methods. For

```
// Create an Apex method to retrieve the recent replies marked as unread in all communitie
public class IdeasController {

    public Idea[] getUnreadRecentReplies() {
        Idea[] recentReplies;
        if (recentReplies == null) {
            Id[] recentRepliesIds = Ideas.getUnreadRecentReplies(UserInfo.getUserId(), '');
            recentReplies = [SELECT Id, Title FROM Idea WHERE Id IN :recentRepliesIds];
        }
        return recentReplies;
    }

}
```

The following is the markup for a Visualforce page that uses the above custom controller to list unread recent replies.

```
<apex:page controller="IdeasController" showHeader="false">
    <apex:dataList value="{!unreadRecentReplies}" var="recentReplyIdea">
        <a href="/apex/viewIdea?id={!recentReplyIdea.Id}">
            <apex:outputText value="{!recentReplyIdea.Title}" escape="true"/></a>
    </apex:dataList>
</apex:page>
```

The following example uses a Visualforce page in conjunction with a custom controller to list ideas. Then, a second Visualforce page and custom controller is used to display a specific idea and mark it as read. For this example to work, there must be ideas posted to the zone.

```
// Create a controller to use on a VisualForce page to list ideas
public class IdeaListController {

    public final Idea[] ideas {get; private set;}

    public IdeaListController() {
        Integer i = 0;
        ideas = new Idea[10];
        for (Idea tmp : Database.query
('SELECT Id, Title FROM Idea WHERE Id != null AND parentIdeaId = null LIMIT 10')) {
            i++;
            ideas.add(tmp);
        }
    }
}
```

The following is the markup for a Visualforce page that uses the above custom controller to list ideas:

```
<apex:page controller="IdeaListController" tabStyle="Idea" showHeader="false">

    <apex:dataList value="{!ideas}" var="idea" id="ideaList">
        <a href="/apex/viewIdea?id={!idea.id}">
<apex:outputText value="{!idea.title}" escape="true"/></a>
    </apex:dataList>

</apex:page>
```

The following example also uses a Visualforce page and custom controller, this time, to display the idea that is selected on the above idea list page. In this example, the `markRead` method marks the selected idea and associated comments as read by the user that is currently logged in. Note that the `markRead` method is in the constructor so that the idea is marked read immediately when the user goes to a page that uses this controller. For this example to work, there must be ideas posted

```
    // Create an Apex method in the controller that marks all comments as read for the
    // selected idea
    public class ViewIdeaController {

        private final String id = System.currentPage().getParameters().get('id');

        public ViewIdeaController(ApexPages.StandardController controller) {
                Ideas.markRead(id);
        }

    }
```

The following is the markup for a Visualforce page that uses the above custom controller to display the idea as read.

```
<apex:page standardController="Idea" extensions="ViewIdeaController" showHeader="false">

    <h2><apex:outputText value="{!idea.title}" /></h2>
    <apex:outputText value="{!idea.body}" />

</apex:page>
```

Ideas Methods

The following are methods for `Ideas`. All methods are static.

- **findSimilar(idea)**
  Returns a list of similar ideas based on the title of the specified idea.

- **getAllRecentReplies(userID, communityID)**
  Returns ideas that have recent replies for the specified user or zone. This includes all read and unread replies.

- **getReadRecentReplies(userID, communityID)**
  Returns ideas that have recent replies marked as read.

- **getUnreadRecentReplies(userID, communityID)**
  Returns ideas that have recent replies marked as unread.

- **markRead(ideaID)**
  Marks all comments as read for the user that is currently logged in.

findSimilar(idea)

Returns a list of similar ideas based on the title of the specified idea.

Signature

```
public static ID[] findSimilar(Idea idea)
```

Parameters

idea
  Type: Idea

Return Value

Type: ID[]

Usage

Each `findSimilar` call counts against the SOSL query limits. See Execution Governors and Limits.

Signature

```
public static ID[] getAllRecentReplies(String userID, String communityID)
```

Parameters

userID

Type: String

communityID

Type: String

Return Value

Type: ID[]

Usage

Each `getAllRecentReplies` call counts against the SOQL query limits. See Execution Governors and Limits.

getReadRecentReplies(userID, communityID)

Returns ideas that have recent replies marked as read.

Signature

```
public static ID[] getReadRecentReplies(String userID, String communityID)
```

Parameters

userID

Type: String

communityID

Type: String

Return Value

Type: ID[]

Usage

Each `getReadRecentReplies` call counts against the SOQL query limits. See Execution Governors and Limits.

getUnreadRecentReplies(userID, communityID)

Returns ideas that have recent replies marked as unread.

Signature

```
public static ID[] getUnreadRecentReplies(String userID, String communityID)
```

Parameters

userID

Type: String

communityID

Type: String

Each `getUnreadRecentReplies` call counts against the SOQL query limits. See [Execution Governors and Limits](#).

markRead(ideaID)

Marks all comments as read for the user that is currently logged in.

Signature

```
public static Void markRead(String ideaID)
```

Parameters

ideaID

Type: String

Return Value

Type: Void

DID THIS ARTICLE SOLVE YOUR ISSUE?
Let us know so we can improve!

[Share your feedback](#)

DEVELOPER CENTERS

Heroku

MuleSoft

Tableau

Commerce Cloud

Lightning Design System

Einstein

Quip

POPULAR RESOURCES

Documentation

Component Library

APIs

Trailhead

Sample Apps

Podcasts

AppExchange

COMMUNITY

Trailblazer Community

Events and Calendar

Partner Community

Blog

Salesforce Admins

Salesforce Architects

Privacy Information   Terms of Service   Legal   Use of Cookies   Trust   Cookie Preferences

Your Privacy Choices   Responsible Disclosure   Contact