Developers   ⌄

OrgPartition Class   ☰

Apex Reference Guide / **Cache Namespace** / OrgPartition Class

OrgPartition Class

Contains methods to manage cache values in the org cache of a specific partition. Unlike the session cache, the org cache is not tied to any session. It's available to the organization across requests and to all users.

Namespace

Cache

Usage

This class extends Cache.Partition and inherits all its non-static methods. Utility methods for creating and validating keys aren't supported and can be called only from the `Cache.Partition` parent class. For a list of `Cache.Partition` methods, see Partition Methods.

To get an org partition, call `Cache.Org.getPartition` and pass in a fully qualified partition name, as follows.

```
Cache.OrgPartition orgPartition = Cache.Org.getPartition('namespace.myPartition');
```

See Cache Key Format for Partition Methods.

Example

This class is the controller for a sample Visualforce page (shown in the subsequent code sample). The controller shows how to use the methods of `Cache.OrgPartition` to manage a cache value on a particular partition. The controller takes inputs from the Visualforce page for the partition name, key name for a counter, and initial counter value. The controller contains default values for these inputs. When you click **Rerender** on the Visualforce page, the `go()` method is invoked and increases the counter by one. When you click **Remove Key**, the counter key is removed from the cache. The counter value gets reset to its initial value when it's re-added to the cache.

> ℹ **Note**
>
> If another user logs in and runs this sample, the user gets the cache values that were last added or updated by the previous user. For example, if the counter value was five, the next user sees the counter value as increased to six.

```apex
public class OrgPartitionController {

    // Name of a partition
    String partitionInput = 'local.myPartition';
    // Name of the key
    String counterKeyInput = 'counter';
    // Key initial value
    Integer counterInitValue = 0;
    // Org partition object
    Cache.OrgPartition orgPartition;
```

```apex
// This method is called when the Visualforce page loads.
public void init() {
    // Create the partition instance based on the partition name
    orgPartition = getPartition();

    // Create the partition instance based on the partition name
    // given in the Visualforce page or the default value.
    orgPartition = Cache.Org.getPartition(partitionInput);

    // Add counter to the cache with an initial value
    //  or increment it if it's already there.
    if (!orgPartition.contains(counterKeyInput)) {
        orgPartition.put(counterKeyInput, counterInitValue);
    } else {
        orgPartition.put(counterKeyInput, getCounter() + 1);
    }
}

// Returns the org partition based on the partition name
// given in the Visualforce page or the default value.
private Cache.OrgPartition getPartition() {
    if (orgPartition == null) {
        orgPartition = Cache.Org.getPartition(partitionInput);
    }

    return orgPartition;
}


// Return counter from the cache.
public Integer getCounter() {
    return (Integer)getPartition().get(counterKeyInput);
}

// Invoked by the Submit button to save input values
//  supplied by the user.
public PageReference save() {
    // Reset the initial key value in the cache
    getPartition().put(counterKeyInput, counterInitValue);

    return null;
}

// Method invoked by the Rerender button on the Visualforce page.
// Updates the values of various cached values.
// Increases the values of counter and the MyData counter if those
//  cache values are still in the cache.
public PageReference go() {
    // Get the org partition object
    orgPartition = getPartition();
    // Increase the cached counter value or set it to 0
    //  if it's not cached.
    if (orgPartition.contains(counterKeyInput)) {
        orgPartition.put(counterKeyInput, getCounter() + 1);
    } else {
        orgPartition.put(counterKeyInput, counterInitValue);
    }

    return null;
}

// Method invoked by the Remove button on the Visualforce page.
// Removes the datetime cached value from the org cache.
public PageReference remove() {
    getPartition().remove(counterKeyInput);

    return null;
}

// Get and set methods for accessing variables
// that correspond to the input text fields on
// the Visualforce page.
```

```
            }

        public Integer getCounterInitValue() {
            return counterInitValue;
        }

        public void setPartitionInput(String partition) {
            this.partitionInput = partition;
        }

        public void setCounterKeyInput(String keyName) {
            this.counterKeyInput = keyName;
        }

        public void setCounterInitValue(Integer counterValue) {
            this.counterInitValue = counterValue;
        }
    }
}
```

This is the Visualforce page that corresponds to the `OrgPartitionController` class.

```
<apex:page controller="OrgPartitionController" action="{!init}">

    <apex:form >
        <br/>Partition with Namespace Prefix: <apex:inputText value="{!partitionInput}"/>
        <br/>Counter Key Name: <apex:inputText value="{!counterKeyInput}"/>
        <br/>Counter Initial Value: <apex:inputText value="{!counterInitValue}"/>
        <apex:commandButton action="{!save}" value="Save Key Input Values"/>
    </apex:form>

    <apex:outputPanel id="output">
        <br/>Cached Counter: <apex:outputText value="{!counter}"/>
    </apex:outputPanel>

    <br/>
    <apex:form >
        <apex:commandButton id="go" action="{!go}" value="Rerender" rerender="output"/>
        <apex:commandButton id="remove" action="{!remove}" value="Remove Key" rerender="ou
    </apex:form>

</apex:page>
```

See Also

- *Apex Developer Guide*: Platform Cache

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

Share your feedback

Commerce Cloud

Lightning Design System

Einstein

Quip

Trailhead

Sample Apps

Podcasts

AppExchange

Blog

Salesforce Admins

Salesforce Architects

Privacy Information

Terms of Service

Legal

Use of Cookies

Trust

Cookie Preferences

Your Privacy Choices

Responsible Disclosure

Contact

Commerce Cloud

Lightning Design System

Trailhead

Sample Apps

Blog

Salesforce Admins