

# Using the MemoryService

## Information

This explains basic usage of the Sakai MemoryService. This service manages and is used to create and retrieve information about existing caches.

- Trunk javadocs:
  - MemoryService: <http://nightly2.sakaiproject.org/javadoc/sakai-javadoc/org/sakaiproject/memory/api/MemoryService.html>
  - Cache: <http://nightly2.sakaiproject.org/javadoc/sakai-javadoc/org/sakaiproject/memory/api/Cache.html>
- Trunk source location: <https://source.sakaiproject.org/svn/kernel/trunk/api/src/main/java/org/sakaiproject/memory/>

## Accessing the MemoryService

- You can use Spring Framework to inject the service or use the cover

1. Using Spring to get the service for your class (e.g. YourAppClass) (*recommended*)
  - a. Add the MemoryService bean to the bean for YourAppClass

```
<bean id="org.sakaiproject.yourapp.logic.YourAppClass"
      class="org.sakaiproject.yourapp.logic.impl.YourAppClassImpl">
  <property name="memoryService"
            ref="org.sakaiproject.memory.api.MemoryService" />
</bean>
```

- b. Add a variable and setter to YourAppClass to use the service in like so:

```
private MemoryService memoryService;
public void setMemoryService(MemoryService memoryService) {
    this.memoryService = memoryService;
}
```

2. Using the Component Manager to get the service
  - **Note:** This is not the recommended method, you should be using Spring to inject the service
    - a. Use the CM cover to get the service

```
import org.sakaiproject.component.cover.ComponentManager;
...
private MemoryService memoryService;
...
memoryService = (MemoryService) ComponentManager.get(MemoryService.class);
```

## Getting a Cache by name

1. Use the service variable to access the service and request a Cache. This will create one if needed or retrieve an existing one if it was created earlier.

```
Cache myCache = memoryService.getCache("org.sakaiproject.my.MyService.myCache");
```

- **Note:** This returns a Sakai Cache object