

# Posting Text for Focus Analysis (since 0.8.0)

The `org.opencode4workspace.endpoints.FocusPostEndpoint` is used to post text for Focus analysis and retrieve a List of Focus objects corresponding to the analyses.

The FocusEndpoint is instantiated, like the other endpoints, by passing in the **WWClient** for the request. The text string is posted using the `postMessage` (*String text*) method. Alternatively, the `WWClient.postTextForFocusAnalysis(String text)` helper method is available for easier use.

A List of Focus objects is returned. The **applicationId** and **lensId** are auto-generated. The **focusVersion** is defined by Watson Work Services. The **lens** identifies what was identified - an "ActionRequest", a "Question" or a "Commitment". The confidence is a number between 0 and 1 that identifies how certain Watson Work Services was in identifying the lens. The **start** and **end** are the indices of within the passed text that the **phrase** can be found. It seems that the lens always has `hidden` set to true.

## postTextForFocus

```
@Test(enabled = true)
@Parameters({ "appId", "appSecret" })
public void postTestForFocus(String appId, String appSecret) throws UnsupportedOperationException, WWException {
    String TEXT = "Hello, I think we need to reconsider. Let's add a meeting on Tuesday 13th March. Frank, what
do you think?";
    WWClient client = WWClient.buildClientApplicationAccess(appId, appSecret, new WWAAuthenticationEndpoint());
    assert !client.isAuthenticated();
    client.authenticate();
    assert client.isAuthenticated();

    List<Focus> focuses = client.postTextForFocusAnalysis(TEXT);
    assert (focuses.size() == 3);
    assert (focuses.get(0).getLens().equals(Lens.ACTION_REQUEST));
    assert (focuses.get(2).getLens().equals(Lens.QUESTION));
    assert (38 == focuses.get(0).getEnd());
    assert (1.0 == focuses.get(1).getConfidence());
}
```