

Use the Bullhorn SOAP API to Work with Notes

Introduction

This tutorial is for developers who create custom applications that use the Bullhorn SOAP-based web services APIs. The tutorial describes how to work with the Note entity in the Bullhorn system. Code samples are provided in the tutorial for C#, Java, and PHP.

You learn how to do the following tasks:

- Add a new note in the Bullhorn system by using the API.
- Query the notes.

Prerequisites

Bullhorn recommends the following prerequisites for this tutorial:

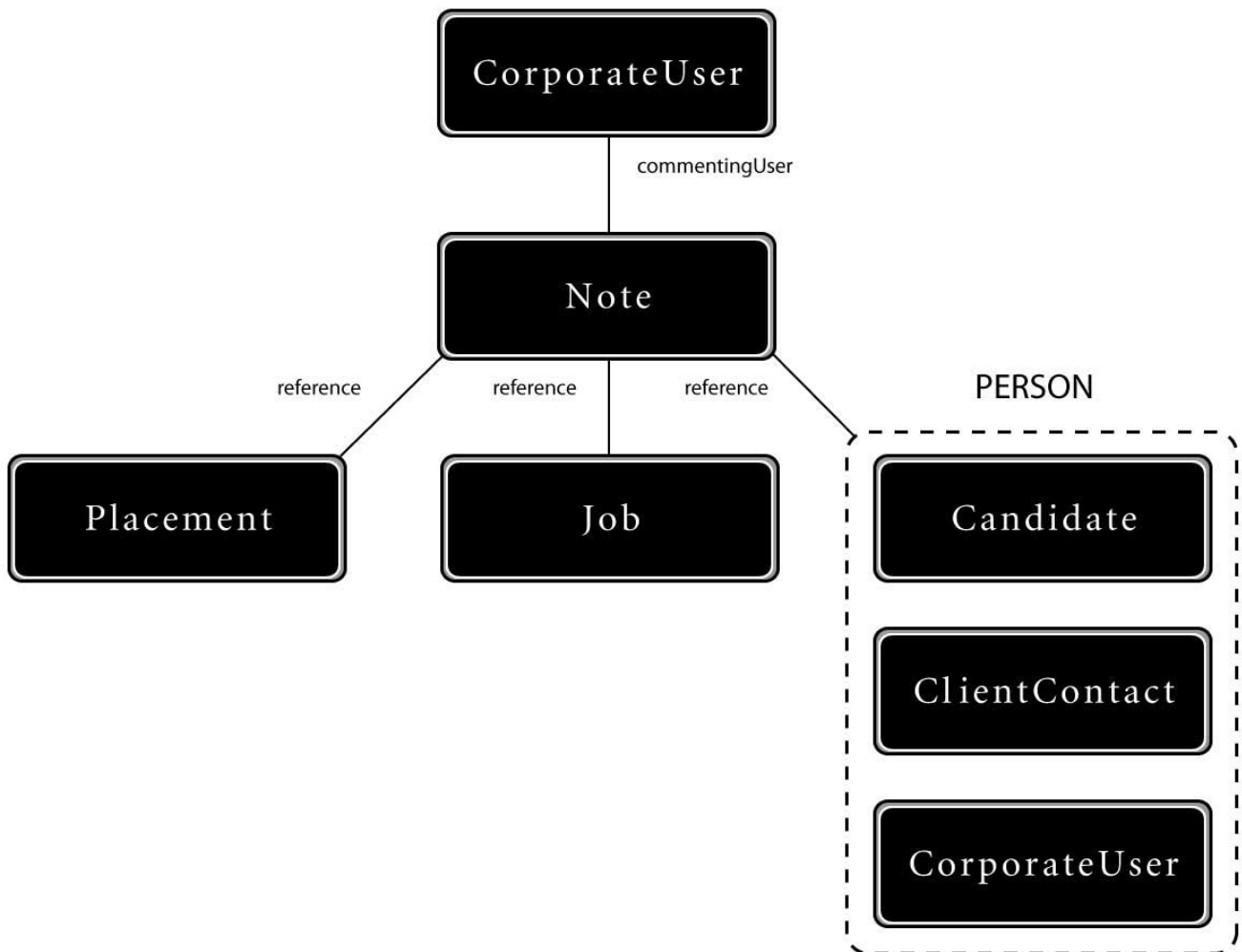
- Knowledge of C#/ASP.NET, Java programming, or PHP
- Knowledge of SOAP and HTML
- It is recommended that you familiarize yourself with one of the Getting Started tutorials

Understanding Notes

The Note entity represents a note (comment) associated with a Candidate, ClientContact, or CorporateUser record. You access notes on the Notes tab on the person's record in the Bullhorn application.

Notes let users record information about their interactions with clients and candidates in the Bullhorn application, so the notes are easily searchable and accessible to others in the company.

The actual data contained in the Note entity is relatively simple. The entity has a comments field that contains text, a commentingUserID reference that names the author, and an action. Use references to establish relationships among notes and entities in the Bullhorn application. References are links that you create between a note and the other entities. A single note can have multiple references, as the following figure shows.



After you enter a note in the Bullhorn application, the note appears in the notes list for any of the entities to which it has a reference. Then a user working with that entity can see the comments or actions that are related to that entity.

Adding a Note DTO

The following sample code creates a note. You create an instance of the Note Data Transfer Object (DTO), set properties on the DTO, and save the note in the database.

Before you create a note, you must have the IDs of some entities to reference. You can set up a query operation to retrieve the IDs of the entities that you will use to create a note, or you can retrieve their IDs from the Bullhorn application.

You perform the following tasks in C#, Java, and PHP:

- Create a session to the Bullhorn web service using your credentials.
- Create an instance of the Note DTO.
- Set the properties in the Note DTO.
- Save the Note DTO.

Creating a session

You must create a session for all calls to the Bullhorn SOAP API. To create a session to the Bullhorn system in the web application, you require a username, password, and API key. For a detailed explanation of creating sessions, see the Getting Started articles.

Java code

```
private static final QName SERVICE_NAME
    = new QName("http://apiservice.bullhorn.com/", "ApiService");
String username = "yourusername";
String password = "yourpassword";
String apiKey = "yourapikey";
String wsdlUrl = "https://api.bullhornstaffing.com/webservices-1.1/?wsdl";
URL serviceUrl = new URL(ApiService_Service.class.getResource("."), wsdlUrl);
ApiService service = new ApiService_Service(serviceUrl, SERVICE_NAME).getApiServicePort();
ApiSession currentSession = service.startSession(username, password, apiKey);
```

C# code

```
String username = "yourusername";
String password = "yourpassword";
String apiKey = "yourapikey";
ApiService service = new ApiService();
apiSession currentSession = new apiSession();
currentSession = service.startSession(username, password, apiKey);
```

PHP code

```
$params = array(
    'trace' => 1,
    'soap_version' => SOAP_1_1);
$BHClient = new SoapClient("https://api.bullhornstaffing.com/webservices-1.1/?wsdl", $params);

$username = "yourusername";
$password = "yourpassword";
$apiKey = "yourapikey";

$session_request = new stdClass();
$session_request->username = $username;
$session_request->password = $password;
$session_request->apiKey = $apiKey;
$API_session = $BHClient->startSession($session_request);
$API_currentSession = $API_session->return;
```

Tip: Refreshing sessions

Sessions have a short expiration and must be refreshed with each call to the Bullhorn SOAP API. Each call to the Bullhorn SOAP API requires a valid session object, and each response returns a new session object. Store that new session object in the current session variable so you always have a recently refreshed value.

Use the session object that is returned as part of the response in the next call that is made to the API, because previous session objects expire in 5 minutes.

Creating an instance of the Note DTO and setting properties

To create a note, you must create an instance of the Note DTO class and set properties for the note. There are multiple properties associated with a note. For the complete list, see the reference documentation.

The following table lists the properties that you set for a note.

Properties	Description
action	The action type associated with the note. The list of values is configured in the private label attribute called commentActionList.
commentingPersonID	The Bullhorn ID of the person who created the note.
personReferenceID	The Bullhorn ID of the person with whom this note is associated. Note: This field must contain a value, but you also must add the reference by using the addNoteReference operation. This lets you create an association with multiple people from a single note.
comments	The text of the note.

Java code

1. Create a note

```
NoteDto noteDto = new NoteDto();
noteDto.setAction("Sample");
noteDto.setCommentingPersonID(currentSession.getUserId());
noteDto.setComments("This is the best note ever.");
```

2. Use the candidate's user Id that you retrieve in the code below. We are defining the variable thisCandidate as an example.

```
noteDto.setPersonReferenceID(thisCandidate.getUserID());
```

C# code

1. Create a note

```
noteDto noteDto = new noteDto();
noteDto.action = "Sample";
noteDto.commentingPersonID = currentSession.userId;
noteDto.commentingPersonIDSpecified = true;
noteDto.comments = "This is the best note ever, coming from C#.";
```

2. Use the candidate's user Id that you retrieve in the code below. We are defining the variable thisCandidate as an example.

```
noteDto.personReferenceID = thisCandidate.userId;
```

```
noteDto.personReferenceIDSpecified = true;
```

PHP code

1. Create a note

```
$note_array = array(  
    'action' => 'Sample',  
    'commentingPersonID' => $API_currentSession->userId,  
    'comments' => 'This is the best note ever, coming from PHP.',
```

2. Use the candidate's user Id that you retrieve in the code below. We are defining the variable `$thisCandidate` as an example.

```
'personReferenceID' => new SoapVar($thisCandidate->userId, XSD_INTEGER, "int",  
    "http://www.w3.org/2001/XMLSchema") //  
);
```

Saving the Note DTO

After you create a note, you must save it to persist the record in the Bullhorn database. Use the save operation to create the Note entity instance and transmit the Note DTO to the Bullhorn server. If the DTO includes a value for the entity's primary key or ID, the entity instance represented by that ID is updated with the values you provide. If the DTO that you send does not include a value for the primary key, a new entity instance is created, and the response message includes the new entity's ID.

Java code

1. Save the Note

```
ApiSaveResult results = service.save(currentSession, noteDto);
```

2. The API returns a new session object. Refresh the session after calling the save operation.

```
currentSession = results.getSession();
```

C# code

1. Save the note

```
apiSaveResult results = service.save(currentSession, noteDto);
```

2. The API returns a new session object. Refresh the session after calling the save operation.

```
currentSession = results.session;
```

PHP code

```
$$SOAP_dto = new SoapVar($note_array, SOAP_ENC_OBJECT, "noteDto",  
    "http://note.entity.bullhorn.com/");
```

```

$request_array = array(
    'session' => $API_currentSession,
    'dto' => $SOAP_dto
);

$SOAP_request = new SoapVar($request_array, SOAP_ENC_OBJECT, "save",
    "http://save.apiservice.bullhorn.com/");

try{
    $saveResponse = $BHClient->save($SOAP_request);
} catch (SoapFault $fault) {
    var_dump($BHClient->__getLastRequest());
    die($fault->faultstring);
}

$API_currentSession = $saveResponse->return->session;
$savedNoteDto = $saveResponse->return->dto;

```

Adding Note References

Use the addNoteReference operation to create links to other entities. The addNoteReference operation requires four arguments: the current session, the ID of the note that you created, the entity name to which you create a reference, and the ID of the entity. The addNoteReference operation returns a response of type addNoteReferenceResult..

Tip: Properties for linking a note

Even though someone is listed in the personReferenceId property, they also must be added as a reference using the addNoteReference operation. For the purposes of data entry, you must include the personReferenceId property, however, the Bullhorn application does not use this property.

Adding a reference to a candidate

To add a reference to a candidate, use the addNoteReference operation, as the following code samples show.

Java code

1. Save a note

```
NoteDto savedNoteDto = (NoteDto)results.getDto();
```

2. Associate the note with a candidate: Use the candidate's user Id that you retrieve in the code below. We are defining the variable thisCandidate as an example.

```
service.addNoteReference(currentSession, savedNoteDto.getNoteID(), "Candidate",
    thisCandidate.getUserID());
```

C# code

1. Save a note

```
noteDto savedNoteDto = (noteDto)results.dto;
```

2. Associate the note with a candidate: Use the candidate's user Id that you retrieve in the code below. We are defining the variable `thisCandidate` as an example.

```
service.addNoteReference(currentSession, SavedNoteDto.noteID, "Candidate",
    thisCandidate.userID);
```

PHP code

1. Use the candidate's user Id that you retrieve in the code below. We are defining the variable `$thisCandidate` as an example.

```
$request_array = array(
    'session' => $API_currentSession,
    'noteId' => new SoapVar($savedNoteDto->noteID, XSD_INTEGER, "int",
        "http://www.w3.org/2001/XMLSchema"),
    'entityName' => 'Candidate',
    'entityId' => new SoapVar($thisCandidate->userID, XSD_INTEGER, "int",
        "http://www.w3.org/2001/XMLSchema")
);
$SOAP_request = new SoapVar($request_array, SOAP_ENC_OBJECT, "addNoteReference",
    "http://apiservice.bullhorn.com/");
try{
    $addNoteReferenceResponse = $BHClient->addNoteReference($SOAP_request);
} catch (SoapFault $fault) {
    var_dump($BHClient->__getLastRequest());
    die($fault->faultstring);
}
$API_currentSession = $addNoteReferenceResponse->return->session;
```

Adding a reference to a job

To add a reference to a job, use the `addNoteReference` operation, as the following code samples show.

Java code

Associate the note with a job order

```
service.addNoteReference(currentSession, SavedNoteDto.getNoteID(), "JobOrder",
    thisJobOrder.getJobOrderID());
```

C# code

Associate the note with a job order Use the Job's Id that you are adding a reference to in the code below. We are defining the variable `thisJobOrder` as an example.

```
service.addNoteReference(currentSession, SavedNoteDto.noteID, "JobOrder",
    thisJobOrder.jobOrderID);
```

PHP code

Use the Job's Id that you are adding a reference to in the code below. We are defining the variable `$thisJobOrder` as an example.

```
$request_array = array(
```

```

'session' => $API_currentSession,
'noteId' => new SoapVar($savedNoteDto->noteID, XSD_INTEGER, "int",
    "http://www.w3.org/2001/XMLSchema"),
'entityName' => 'JobOrder',
'entityId' => new SoapVar($thisJobOrder->jobOrderID, XSD_INTEGER, "int",
    "http://www.w3.org/2001/XMLSchema") //
);
$SOAP_request = new SoapVar($request_array, SOAP_ENC_OBJECT, "addNoteReference",
    "http://apiservice.bullhorn.com/");
try{
    $addNoteReferenceResponse = $BHClient->addNoteReference($SOAP_request);
} catch (SoapFault $fault) {
    var_dump($BHClient->__getLastRequest());
    die($fault->faultstring);
}
$API_currentSession = $addNoteReferenceResponse->return->session;

```

Removing a note reference

To remove note references, use the `removeNoteReference` operation. The `removeNoteReference` operation requires four arguments: the current session, the ID of the note that you remove, the entity name to which you are removing a reference, and the ID of the entity.

Java code

Retrieve the client contact id before running the code below. We are defining the variable `thisClientContact` as an example.

```

service.removeNoteReference(currentSession, SavedNoteDto.getNoteID(), "ClientContact",
    thisClientContact.getUserID());

```

C# code

Retrieve the client contact id before running the code below. We are defining the variable `thisClientContact` as an example.

```

service.removeNoteReference (currentSession, SavedNoteDto.noteID, "ClientContact",
    thisClientContact.userID);

```

PHP code

Retrieve the client contact id before running the code below. We are defining the variable `thisClientContact` as an example.

```

$request_array = array(
    'session' => $API_currentSession,
    'noteId' => new SoapVar($savedNoteDto->noteID, XSD_INTEGER, "int",
        "http://www.w3.org/2001/XMLSchema"),
    'entityName' => 'ClientContact',
    'entityId' => new SoapVar($thisClientContact->userID, XSD_INTEGER, "int",
        "http://www.w3.org/2001/XMLSchema")
);

$SOAP_request = new SoapVar($request_array, SOAP_ENC_OBJECT, "removeNoteReference",
    "http://apiservice.bullhorn.com/");
try{
    $removeNoteReferenceResponse = $BHClient->removeNoteReference($SOAP_request);
} catch (SoapFault $fault) {
    var_dump($BHClient->__getLastRequest());
}

```



```
die($fault->faultstring);
}
```

Querying a Note

Several operations let you retrieve notes from the Bullhorn application. You query a note using the `getEntityNotes` operation and the `getEntityNotesWhere` operation.

Using the `getEntityNotes` operation

The `getEntityNotes` operation retrieves the IDs for all note instances associated with a `Candidate`, `ClientContact`, `CorporateUser`, `JobOrder`, or `Placement` entity. The operation returns a response of type `ApiGetEntityNotesResult`. The response contains two parameters: the currently active session, and a list of `Note` instance IDs that match the query.

To retrieve individual `Note` DTOs, use the `find` operation with the ID from the `getEntityNotes` operation.

Java code

1. The candidate ID '6521' below is used as an example. Please use the ID of the candidate on your account.

```
ApiGetEntityNotesResult resultNotes = service.getEntityNotes(currentSession,
    "Candidate", 6521);
```

2. Refresh the session

```
currentSession = resultNotes.getSession();
```

3. Retrieve the Notes

```
for (int i = 0; i < resultNotes.getIds().size(); i++) { ApiFindResult findNoteResult =
    service.find(currentSession, "Note", resultNotes.getIds().get(i));
```

4. Refresh the session

```
currentSession = findNoteResult.getSession();
```

5. Get the Note DTOs

```
NoteDto thisNote = (NoteDto) findNoteResult.getDto();
```

6. Print the note's comment field to console

```
System.out.println("Found note using getEntityNotes() operation:
    " + thisNote.getComments());
}
```

C# code

1. The candidate ID '6521' below is used as an example. Please use the ID of the candidate on your account.

```
apiGetEntityNotesResult resultNotes = service.getEntityNotes(currentSession,
    "Candidate", 6521);
```

2. Refresh the session

```
currentSession = resultNotes.session;
```

3. Retrieve the Notes

```
foreach (int i in resultNotes.ids)
{
    apiFindResult findNoteResult = service.find(currentSession, "Note", i);
}
```

4. Refresh the session

```
currentSession = findNoteResult.session;
```

5. Get the Note DTOs

```
noteDto thisNote = (noteDto)findNoteResult.dto;
```

- Run the debugger to see the note instance and the properties on it.

PHP code

The candidate ID '6710' below is used as an example. Please use the ID of the candidate on your account.

```
$request_array = array(
    'session' => $API_currentSession,
    'entityName' => 'Candidate',
    'id' => new SoapVar(6710, XSD_INTEGER, "int", "http://www.w3.org/2001/XMLSchema")
);
$SOAP_request = new SoapVar($request_array, SOAP_ENC_OBJECT, "getEntityNotes",
    "http://apiservice.bullhorn.com/");
try{
    $getEntityNotesResponse = $BHClient->getEntityNotes($SOAP_request);
} catch (SoapFault $fault) {
    var_dump($BHClient->__getLastRequest());
    die($fault->faultstring);
}
$API_currentSession = $getEntityNotesResponse->return->session;
foreach ($getEntityNotesResponse->return->ids as $value){
    $findId = new SoapVar($value, XSD_INTEGER, "int", "http://www.w3.org/2001/XMLSchema");
    $find_request = array (
```

```

'session' => $API_currentSession,
'entityName' => 'Note',
'id' => $findId
);
try{
  $findResult = $BHClient->find($find_request);
} catch (SoapFault $fault) {
  echo $BHClient->__getLastResult();
  die($fault->faultstring);
}
$API_currentSession = $findResult->return->session;
$thisNote = $findResult->return->dto;

echo('Found Note using find(): '.$thisNote->comments.'  
<br>');
}

```

Using the getEntityNotesWhere operation

The getEntityNotesWhere operation retrieves the IDs for all note instances that are associated with a Candidate, ClientContact, CorporateUser, JobOrder, or Placement instance and that meet the criteria specified in the notesWhere property. This is useful if you are looking only for notes that have a particular action or that contain particular words. The operation returns a response of type ApiGetEntityNotesResult. The response contains two parameters: the currently active session and a list of the Note instance IDs that match the query.

To retrieve individual Note DTOs, use the find operation with the ID from the getEntityNotesWhere operation.

Note: Prepend any property that you specify in the notesWhere parameter of the getEntityNotesWhere operation with note.; for example, note.action refers to the action column of the note.

Java code

1. The candidate ID '6521' below is used as an example. Please use the ID of the candidate on your account.

```

ApiGetEntityNotesResult resultNotes = service.getEntityNotesWhere(currentSession,
    "Candidate",6521, "note.action = 'sample'");

```

2. Refresh the session

```

currentSession = resultNotes.getSession();

```

3. Get the Notes

```

for (int i = 0; i < resultNotes.getIds().size(); i++ )
{
  ApiFindResult findNoteResult = service.find(currentSession, "Note",
    resultNotes.getIds().get(i));
}

```

4. Refresh the session

```

currentSession = findNoteResult.getSession();

```

5. Get the Note DTOs

```

NoteDto thisNote = (NoteDto)findNoteResult.getDto();

```

6. Print the note's comment property to console

```
System.out.println("Found note using getEntityNotesWhere() operation:
    " +thisNote.getComments());
}
```

C# code

1. The candidate ID '6521' below is used as an example. Please use the ID of the candidate on your account.

```
apiGetEntityNotesResult resultNotes = service.getEntityNotesWhere(currentSession,
    "Candidate", 6521, "note.action = 'sample'");
```

2. Refresh the session

```
currentSession = resultNotes.session;
```

3. Retrieve the Notes

```
foreach (int i in resultNotes.ids)
{
    apiFindResult findNoteResult =
    service.find(currentSession, "Note", i);
}
```

4. Refresh the session

```
currentSession = findNoteResult.session;
```

5. Get the Note DTOs

```
noteDto thisNote = (noteDto)findNoteResult.dto;
```

6. Run the debugger to see the note instance and the properties on it. */

PHP code

1. The candidate ID '6710' below is used as an example. Please use the ID of the candidate on your account.

```
$request_array = array(
    'session' => $API_currentSession,
    'entityName' => 'Candidate',
    'id' => new SoapVar(6710, XSD_INTEGER, "int", "http://www.w3.org/2001/XMLSchema"),
    'notesWhere' => "note.action = 'sample'"
);

$SOAP_request = new SoapVar($request_array, SOAP_ENC_OBJECT,
"getEntityNotesWhere",
    "http://apiservice.bullhorn.com/");
```

```
try{
    $getEntityNotesWhereResponse = $BHClient->getEntityNotesWhere($SOAP_request);
} catch (SoapFault $fault) {
    var_dump($BHClient->__getLastRequest());
    die($fault->faultstring);
}
```

2. Refresh the session

```
$API_currentSession = $getEntityNotesWhereResponse->return->session;
```

3. Retrieve the Notes

```
foreach ($getEntityNotesWhereResponse->return->ids as $value){
    $findId = new SoapVar($value, XSD_INTEGER,
        "int", "http://www.w3.org/2001/XMLSchema");

    $find_request = array (
        'session' => $API_currentSession,
        'entityName' => 'Note',
        'id' => $findId
    );

    try{
        $findR
        result = $BHClient->find($find_request);
    }catch (SoapFault $fault) {
        echo $BHClient->__getLastResult();
        die($fault->faultstring);
    }

    $API_currentSession = $findResult->return->session;
    $thisNote = $findResult->return->dto;

    echo('Found Note with action of sample using find(): '.$thisNote-
        >comments.'

---


```