

GETTING STARTED

Tutorials

This tutorial will show you how to do a few common CRM operations using PHP code to call the API.

Shared Code

All of the code below will rely on this function to call the API:

```
function CallLacrmApi(string $Function, array $Parameters)
{
    $CurlHandle = curl_init("https://api.lessannoying.com");
    $Headers = array(
        "Content-Type: application/json",
        "Authorization: YOUR_API_KEY_HERE"
    );
    $Body = array(
        "Function" => $Function,
```

```

        "Parameters" => $Parameters
    );

    curl_setopt($CurlHandle, CURLOPT_POSTFIELDS, json_encode($Parameters));
    curl_setopt($CurlHandle, CURLOPT_HTTPHEADER, $Headers);
    curl_setopt($CurlHandle, CURLOPT_RETURNTRANSFER, true);

    $CurlResult = curl_exec($CurlHandle);
    $ReturnValue = false;
    if (curl_errno($CurlHandle)) {
        error_log(curl_error($CurlHandle));
    } else {
        $ResultObject = json_decode($CurlResult, true);
        $HttpCode = curl_getinfo($CurlHandle, CURLINFO_HTTP_CODE);
        if ($HttpCode === 400) {
            error_log("Error $ResultObject[ErrorCode]: " . $ResultObject[ErrorMessage]);
        }
    }
}

curl_close($CurlHandle);
return $ReturnValue;
}

```



Less Annoying CRM

Contact us

Sign up

Log in

API DOCUMENTATION

GETTING STARTED

API Introduction

Ways to connect

Data types

Debugging

```

    }
}

curl_close($CurlHandle);
return $ReturnValue;
}

```

Debugging

Other tips

Tutorials

Shared Code

Adding contacts and
companies

Creating a simple webform

Checking for duplicate
contacts

CORE API FUNCTIONS

Contacts / Companies

Items attached to contacts

Events

Group Memberships

Notes

Pipeline Items

Adding contacts and companies

Create two contacts in the CRM: a company, and a contact that works at that company.

This is done by creating a contact with a company name that matches an existing company in the CRM. If you create a contact with a company name that is not in the CRM, a new company with that name will also be created.

Note that duplicate company names are not allowed in the CRM and will cause errors.

```
// First, get the user ID to assign the contacts to.
$UserResults = CallLacrmApi("GetUser");
$UserId = $UserResults["UserId"];

// Second, create a company.
$CompanyName = "Less Annoying CRM";
CallLacrmApi("CreateContact", array(
    "IsCompany" => true,
    "AssignedTo" => $UserId,
    "Company Name" => $CompanyName
));
```

```
// Third, create a contact at the company by matching  
CallLacrmApi("CreateContact", array(  
    "IsCompany" => false,  
    "AssignedTo" => $UserId,  
    "Name" => "Test Contact",  
    "Company Name" => $CompanyName  
));
```

Creating a simple webform

This code displays a simple form to the user for contact information, as well as a field for a comment. Submitting the form creates a contact in the CRM and adds the comment as a note on that contact.

```
if ($_SERVER["REQUEST_METHOD"] === "GET"){  
    echo "  
        Please leave your contact info, and someone wi  
        <form action="" method='post'>  
            <label for='Name'>Name</label>  
            <input type='text' name='Name' id='Name' r  
            <br/>  
        </form>  
    ";  
}
```

```

        <label for="Email">Email</label>
        <input type="email" name="Email" id="Email"
        <br/>
        <label for="Phone">Phone</label>
        <input type="text" name="Phone" id="Phone"
        <br/>
        <label for="Comment">Comment</label>
        <input type="text" name="Comment" id="Comm
        <br/>
        <input type="submit" value="Submit">
    </form>";
} else {
    $UserResults = CallLacrmApi("GetUser");
    $UserId = $UserResults["UserId"];
    $ContactResults = CallLacrmApi("CreateContact", array(
        "IsCompany" => false,
        "AssignedTo" => $UserId,
        "Name" => $_POST["Name"],
        "Email" => array(array("Email" => $_POST["Email"],
        "Phone" => array(array("Phone" => $_POST["Phone"],
    ));
    $ContactId = $ContactResults["ContactId"];
    CallLacrmApi("CreateNote", array("ContactId" => $ContactId));

    echo "Contact form submitted!";
}

```

Checking for duplicate contacts

We can extend the webform example by using GetContacts to check for an existing matching contact before we create the note. This example just checks the email address, but you could also use phone or name.

This code replaces the "else" block from the above example.

```
$Email = trim($_POST["Email"]);

// Search for a contact with this primary email.
$ApiResponse = CallLacrmApi("GetContacts", array(
    "SearchTerms" => $Email
));
$Results = $ApiResponse["Results"];
$ContactId = null;
foreach($Results as $Contact) {
    $ContactEmail = $Contact["Email"][0]["Text"];
    if ($Email === $ContactEmail) {
        $ContactId = $Contact["ContactId"];
        break;
    }
}

// Otherwise, go ahead and make a new contact.
```

```

if (is_null($ContactId)) {
    $UserResults = CallLacrmApi("GetUser");
    $UserId = $UserResults["UserId"];
    $ContactResults = CallLacrmApi("CreateContact", array(
        "IsCompany" => false,
        "AssignedTo" => $UserId,
        "Name" => $_POST["Name"],
        "Email" => array(array("Email" => $_POST["Email"])),
        "Phone" => array(array("Phone" => $_POST["Phone"])),
    ));
    $ContactId = $ContactResults["ContactId"];
}

CallLacrmApi("CreateNote", array("ContactId" => $ContactId));
echo "Contact form submitted!";

```

Next page: [Contacts / Companies](#)