

First Example

```
<template>

    <div class="sldx-box" style="background-color:bisque">

<!--<lightning-record-form object-api-name="Account" record-
id="0015i000009ykf5AAA" layout-type="Full">

</lightning-record-form>-->

<!--Record is pass by user-->

<!--<lightning-record-form object-api-name="Account" record-id={recordId}
layout-type="Full">

</lightning-record-form>-->

<!--recordid is dynamic -->

<!--<lightning-record-form object-api-name={objectApiName} record-
id={recordId} layout-type="Full">

</lightning-record-form>-->

<!--Object api name take dynamically-->

<!--<lightning-record-form object-api-name={objectApiName} record-
id={recordId} layout-type="Full" mode="readonly">

</lightning-record-form>-->

<!--in this example user have to change a mode,mode can be change in
record form>-->

<!--user want to see specific fields-->

<lightning-record-form object-api-name={objectApiName} record-
id={recordId} fields={fields} columns="2">

</lightning-record-form>

<!--user can assign its own field name using fields property-->
```

```
        </div>
    </template>
```

JSFILE

```
import { LightningElement,api} from 'lwc';
import Account_field from '@salesforce/schema/Contact.AccountId';
import Name_field from '@salesforce/schema/Contact.Name';
import email_feild from '@salesforce/schema/Contact.Email';

export default class Recordform extends LightningElement {
    @api recordId;
    @api objectApiName;

    //fields=['AccountId','Name','Type','Rating'];
    //fields=[Account_field,Name_field,email_feild];
}
```

Second Example

```
<template>

    <!--- <lightning-record-view-form object-api-name={objectApiName}
    record-id={recordId}>

        <div class="slds-grid">
            <div class="slds-col slds-size_1-of-2">
                <lightning-output-field field-name="Name"></lightning-
output-field>
```

```

        <lightning-output-field field-name="Phone"></lightning-
output-field>

        </div>

        <div class="slds-col slds-size_1-of-2">

            <lightning-output-field field-name="Industry"></lightning-
output-field>

            <lightning-output-field field-
name="AnnualRevenue"></lightning-output-field>

        </div>

    </div>

</lightning-record-view-form>-->

```

```

<lightning-record-form object-api-name={objectName} fields={fields}
columns="2" onSuccess={handleaccountcreated}>

```

```

</lightning-record-form>

```

```

import { LightningElement, api } from 'lwc';
import { ShowToastEvent } from 'lightning/platformShowToastEvent';

export default class Recordform extends LightningElement {
    @api recordId;

    // @api objectApiName;

    objectName = 'Account';

    fields = ['AccountId', 'Name', 'Type', 'Rating'];

    // fields = [Account_field, Name_field, email_feild];

    handleaccountcreated(event) {

        // here i want to show a toast

        const evt = new ShowToastEvent({

```

```
        title: 'Account sucessfully created',  
        message: 'Account created with this Id'+event.detail.Id,  
        variant: 'success'  
    });  
    this.dispatchEvent(evt);  
}  
}
```

<https://trailhead.salesforce.com/content/learn/modules/lightning-web-components-and-salesforce-data/use-lightning-data-service-to-work-with-data>

Complete trailhead

Complete login form on your project

Complete recordid