Superpowers for Lightning Web Components:
User Interface API

Christian Menzinger
Independent Salesforce Developer & Trainer
christian@menzinger.info
@chris_menzinger

"Please allow me to introduce myself"
Sympathy for the Devil - Rolling Stones

Independent Salesforce Developer & Trainer

5+ years Salesforce addicted   **salesforce** developers

Munich Developer Group Co-Leader   **salesforce** **Trailblazer** COMMUNITY GROUPS

GROUP LEADER

Salesforce MVP   **salesforce** MVP

Salesforce Certified

SALESFORCE CERTIFIED	SALESFORCE CERTIFIED	SALESFORCE CERTIFIED
Administrator	Platform App Builder	Platform Developer I

Christian Menzinger

@chris_menzinger

ChristianMenzinger

What is Lightning?
...not a page based design as Salesforce Classic was

What is Lightning?

When we decided to rebuild our UI,

we didn't reskin it,

we built a powerful platform

that gives you even more access to customize Salesforce.

That's Lightning.

What is Lightning?

user experience
+
UI framework
+
platform services
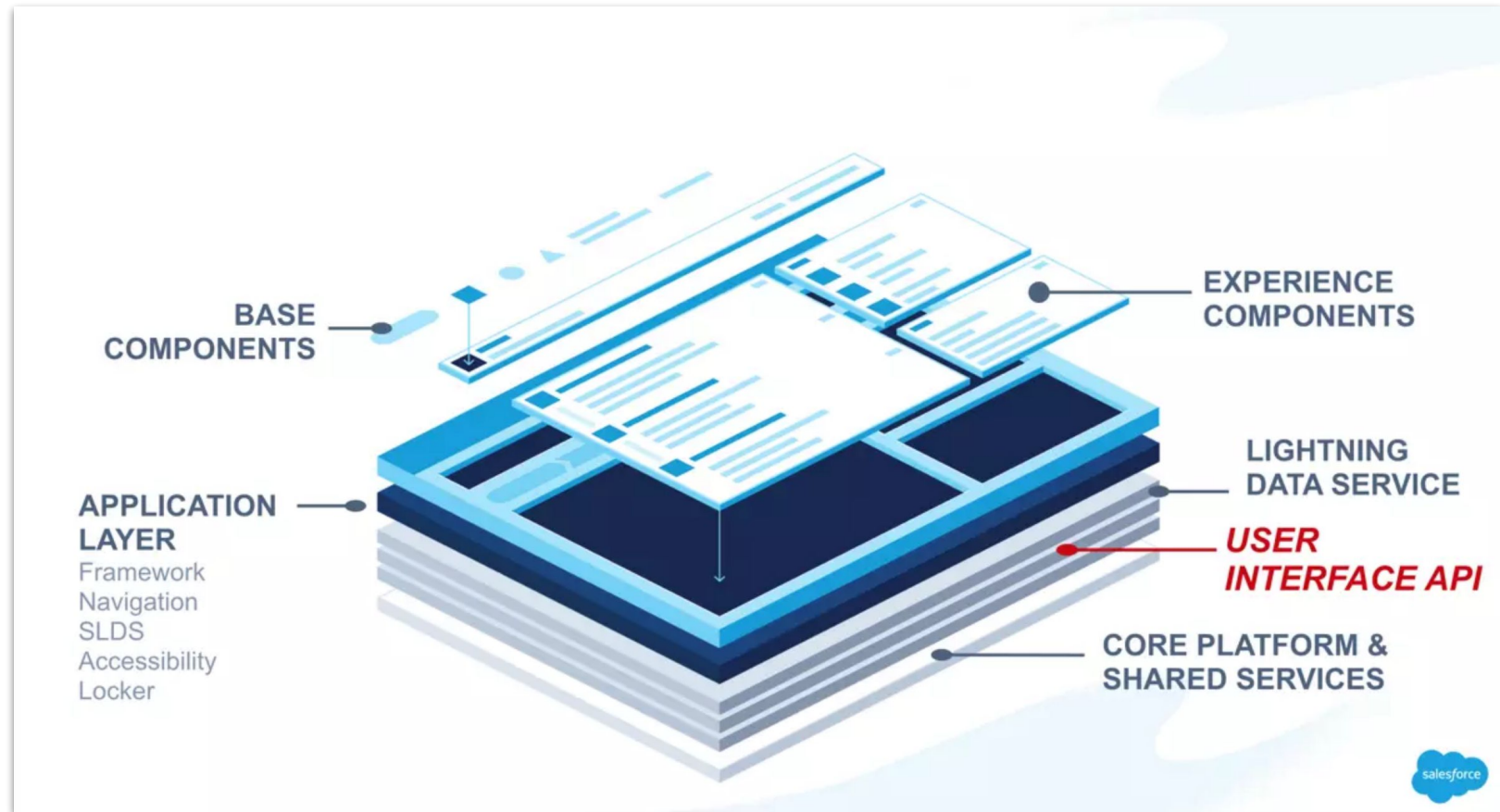
User Interface API

public REST API which returns data and metadata

supports requests with multiple record Ids

designed to build custom applications like iOS apps

hooks for record data, list views, favorites, actions & more

responses respects:

- object permissions (CRUD)

- field level security (FSL)

- sharing settings

User Interface API

Example request

```
1 | GET /ui-api/record-ui/001R0000003GeJ1IAK
```

User Interface API

Example response

layout information

```
017        "layouts":{
018            "Account":{
019                "012000000000000AAA":{
020                    "Full":{
021                        "View":{
022                            "id":"00hR0000000NurZIAS",
023                            "layoutType":"Full",
024                            "mode":"View",
025                            "sections":[
026                                {
027                                    "collapsible":false,
028                                    "columns":2,
029                                    "heading":"Account Information",
030                                    "id":"01BR0000000UjAhMAK",
031                                    "layoutRows":[
032                                        {
033                                            "layoutItems":[
034                                                {
035                                                    "editableForNew":false,
036                                                    "editableForUpdate":false,
037                                                    "label":"Account Owner",
038                                                    "layoutComponents":[
039                                                        {
040                                                            "apiName":"OwnerId",
041                                                            "componentType":"Field",
042                                                            "label":"Owner ID"
043                                                        }
044                                                    ],
045                                                    "lookupIdApiName":"OwnerId",
046                                                    "required":false,
047                                                    "sortable":false
048                                                },
049                                                {
050                                                    "editableForNew":true,
051                                                    "editableForUpdate":true,
052                                                    "label":"Phone",
053                                                    "layoutComponents":[
054                                                        {
055                                                            "apiName":"Phone",
056                                                            "componentType":"Field",
057                                                            "label":"Account Phone"
058                                                        }
```

User Interface API

Example response

object information

```
228        "objectInfos":{
229            "Account":{
230                "apiName":"Account",
231                "childRelationships":[
232                    {
233                        "childObjectApiName":"Account",
234                        "fieldName":"ParentId",
235                        "junctionIdListNames":[

237                        ],
238                        "junctionReferenceTo":[

240                        ],
241                        "relationshipName":"ChildAccounts"
242                    },
243                     ... some child relationships removed for space ...
244                    {
245                        "childObjectApiName":"WorkOrder",
246                        "fieldName":"AccountId",
247                        "junctionIdListNames":[

249                        ],
250                        "junctionReferenceTo":[

252                        ],
253                        "relationshipName":"WorkOrders"
254                    }
255                ],
256                "createable":true,
257                "custom":false,
258                "defaultRecordTypeId":null,
259                "deleteable":true,
260                "feedEnabled":true,
```

User Interface API

Example response

object information

```
261          "fields":{
262              "AccountSource":{
263                  "apiName":"AccountSource",
264                  "calculated":false,
265                  "compound":false,
266                  "compoundComponentName":null,
267                  "compoundFieldName":null,
268                  "controllerName":null,
269                  "createeable":true,
270                  "custom":false,
271                  "dataType":"Picklist",
272                  "extraTypeInfo":null,
273                  "filterable":true,
274                  "filteredLookupInfo":null,
275                  "highScaleNumber":false,
276                  "htmlFormatted":false,
277                  "inlineHelpText":null,
278                  "label":"Account Source",
279                  "length":40,
280                  "nameField":false,
281                  "polymorphicForeignKey":false,
282                  "precision":0,
283                  "reference":false,
284                  "referenceTargetField":null,
285                  "referenceToInfos":[
286
287                  ],
288                  "relationshipName":null,
289                  "required":false,
290                  "scale":0,
291                  "searchPrefilterable":false,
292                  "sortable":true,
293                  "unique":false,
294                  "updateable":true
295              },
```

User Interface API

Example response

record information

```
482    "records":{
483        "001R0000003ILXdIAO":{
484            "apiName":"Account",
485            "childRelationships":{
486
487            },
488            "fields":{
489                "AnnualRevenue":{
490                    "displayValue":null,
491                    "value":null
492                },
493                 ... fields removed for space ...
494                "Website":{
495                    "displayValue":null,
496                    "value":null
497                }
498            },
499            "id":"001R0000003ILXdIAO",
500            "recordTypeInfo":null
501        }
502    }
503 }
```

Implementations of the User Interface API
Lightning Data Service (LDS) & Base Lightning Components

LDS adds caching, synchronization and error handling behaviors

ground layer for Base Lightning Components

easiest way to use the power of the User Interface API:

`<lightning-record-form>` out-of-the-box form with support of view, edit, read-only
for compact or full page layout or specific fields

`<lightning-record-edit-form>` displays an editable form with a custom layout of fields

`<lightning-record-view-form>` displays a read-only form with a custom layout of fields

whenever possible use Base Lightning Components!

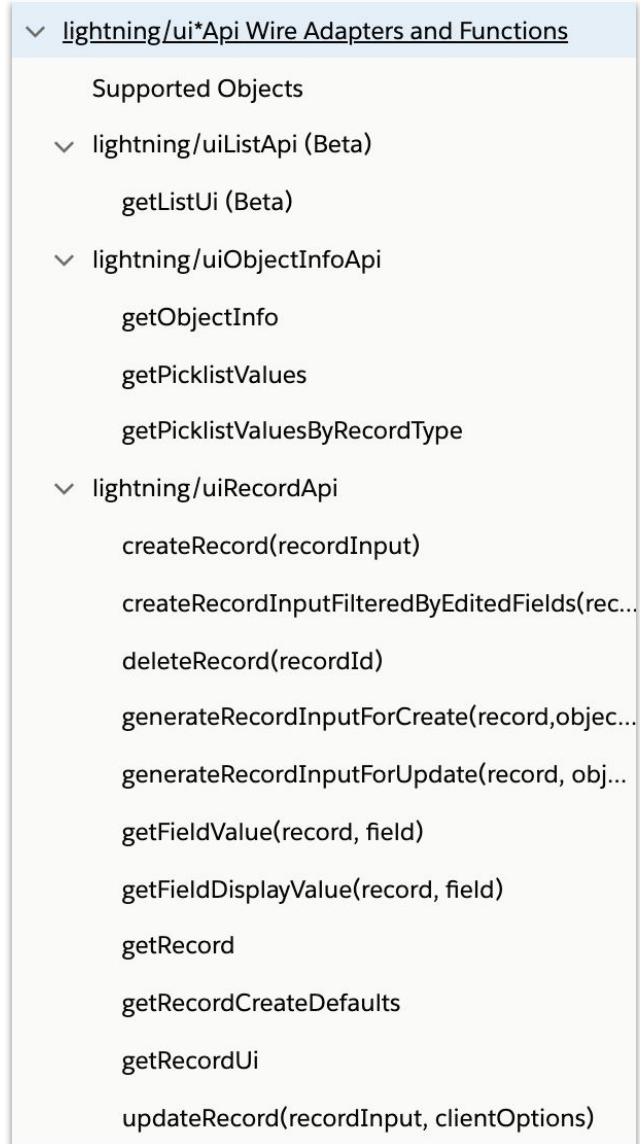# Using the User Interface API with Lightning Web Components
Prebuilt hooks

use the `@wire` adapter to access User Interface API hooks:

```
import { adapterId } from 'adapterModule';

@wire(adapterId, adapterConfig)
propertyOrFunction;
```

developer documentation:
https://developer.salesforce.com/docs/component-library/documentation/lwc/lwc.reference_ui_api

lightning/ui*Api Wire Adapters and Functions

- Supported Objects
- lightning/uiListApi (Beta)
  - getListUi (Beta)
- lightning/uiObjectInfoApi
  - getObjectInfo
  - getPicklistValues
  - getPicklistValuesByRecordType
- lightning/uiRecordApi
  - createRecord(recordInput)
  - createRecordInputFilteredByEditedFields(rec...
  - deleteRecord(recordId)
  - generateRecordInputForCreate(record,objec...
  - generateRecordInputForUpdate(record, obj...
  - getFieldValue(record, field)
  - getFieldDisplayValue(record, field)
  - getRecord
  - getRecordCreateDefaults
  - getRecordUi
  - updateRecord(recordInput, clientOptions)

Lightning Web Components

more standards, less framework

designed to build

complex & sophisticated

components & single page applications

Security Lightning Data Service Base Lightning Components	Lightning Web Components
Web Components Templates Custom elements Shadow DOM Modules ECMAScript 7 Events Standard Elements Rendering	Web Standards

Demo: build an own visualization of a List View
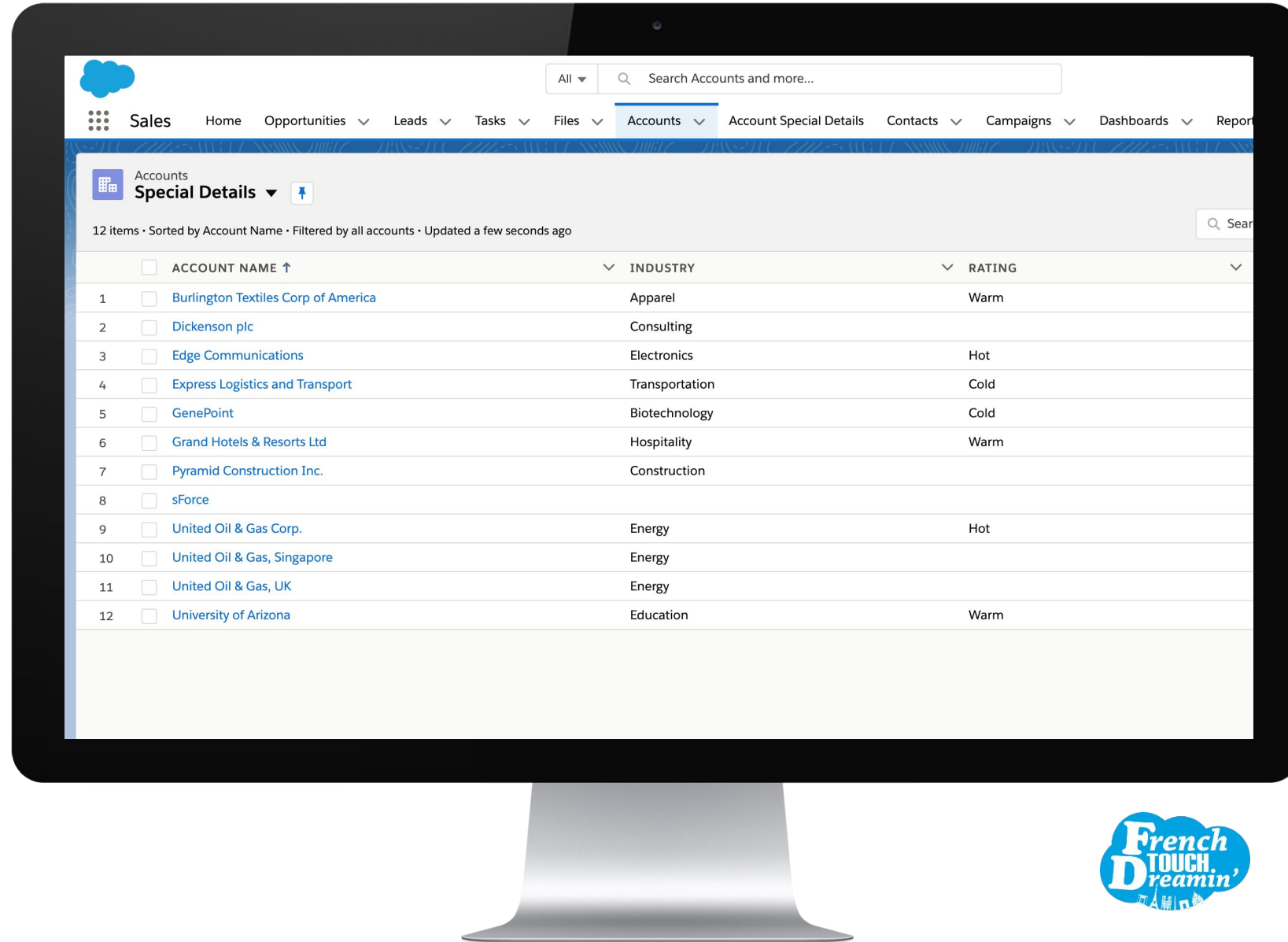
Conclusion

review the prebuilt Base Lightning Components when working with data

review the prebuilt hooks into User Interface API

review sample app for a starting point (NOTE: getListUI is in beta currently):

https://github.com/ChristianMenzinger/FTD19_LWC_UI-API_Demo

Build sophisticated interfaces to thrill your users!

Thank You / Q & A

Remember to tell us what you think in the event survey
www.frenchtouchdreamin.com/survey

Christian Menzinger
Independent Salesforce Developer & Trainer
christian@menzinger.info
@chris_menzinger