

o-spreadsheet: develop custom functions to connect to your database or an external API

Lucas Lefèvre, Software developer



Documents Documents Configuration Documents Pipeline Revenue Report (Monthly) File Edit View Insert Format Data F つ 下 、 % .0 .00 123 * 10 * B I S A → 田 記 三・ ± * | ・ ▼ fx B C D Monthly Revenue by Team - 2023 Actuals Target Forecasted January 2023 100.000.00 0.00% 50.000.00 Sales 0.00% 50,000.00 February 2023 100,000.00 0.00% 0.00 50,000.00 0.00% Pre-Sales 50,000.00 0.00% March 2023 100,000.00 0.00% 0.00 50,000.00 0.00% 12 Pre-Sales 50.000.00 0.00% 13 April 2023 100.000.00 0.00% 0.00 14 Sales 50,000.00 0.00% 15 Pre-Sales 50,000.00 0.00% 16 May 2023 100.000.00 0.00% Sales 50,000.00 0.00% 18 Pre-Sales 50,000.00 0.00% 19 June 2023 100,000.00 0.00% 0.00 50,000.00 0.00% 21 Pre-Sales 50,000,00 0.00% 22 July 2023 100,000.00 0.00% 0.00 50,000.00 24 Pre-Sales 50.000.00 0.00% 25 August 2023 100,000,00 0.00% 0.00 26 Sales 50,000.00 0.00% 27 Pre-Sales 50,000.00 0.00% 28 September 2023 100.000.00 0.00% 0.00 29 Sales 50,000.00 0.00% 30 Pre-Sales 50,000.00 0.00% October 2023 100,000.00 0.00% 0.00 32 Sales 50,000.00 0.00% 50.000.00 Revenue by Team + Revenue by Salesperson → Targets →

On the go with spreadsheets





- 1 Introduction
- 2 Simple custom functions
- 3 Function reading external data
- 4 Performance pitfalls
- 5 Conclusion

o-spreadsheet

- https://github.com/odoo/o-spreadsheet
 - Standalone library written in Typescript Ts
 - Open source
 - Integrated to Odoo
 - Odoo is an external system from o-spreadsheet POV
 - Extensible with plugins and custom functions

What we are building today



=ODOO.SURVEY(1)									
ì	н	I.	J	К	L	М	N	0	Р
	Name	Birthday	Rating	Feedback	City	Country	How frequently	do you buy pr	oducts online?
	Martin Cohps	1979-07-04	4	Great	Wierde	Skipped	Once a year		
	Agnes Monica	1960-02-01	3	Good to know. 1	York	UK	Once a year		
	Julia Ray	1999-06-14	4	See you next tin	Buenos Aires	Argentina	Once a month		
	Karim West	2003-05-04	5	Reaaaaally goo	Tunis	Tunisie	Once a day		
	Loïc Rousseau	1992-08-15	4	I love sailing	Le Palais	France	Less than once	a year	
	Clara Tegas	1960-10-17	5	Very welcoming	Wellington	New Zealand	Once a year		
	Mariah Geury	1973-12-07	2	Not so pleased	Abidjan	Côte d'Ivoire	Once a week		
	Lucie Allard	1993-07-12	2	Thanks!	Bangkok	Thailand	Once a week		
	Marion Cousser	1998-01-04	4	Great! Congrats	Mexico	Mexico	Once a month		
	Clément Ouassi	2000-06-21	3	Not much to say	New York	USA	Once a day		
	Hugo Domken	1990-02-12	5	Excellent. Best	Madrid	Spain	Once a week		
	Antoine Stampe	1975-03-09	4	I liked it	Brussels	Skipped	Once a month		

Declare a function

Declare the arguments

```
import { helpers, registries } from "@odoo/o-spreadsheet";
import { t } from "@web/core/l10n/translation";
const { arg, toNumber } = helpers;
const MY ADD = {
 description: t("Adds two numbers together."),
 args: [
   arg("value1 (number)", t("The first number")),
   arg("value2 (number)", t("The second number")),
  returns: ["NUMBER"],
```

Declare a function

- Declare the arguments
- Implement compute
 - return a primitive value
 - return a 2d array of primitives

```
import { helpers, registries } from "@odoo/o-spreadsheet";
import { t } from "@web/core/110n/translation";
const { arg, toNumber } = helpers;
const MY ADD = {
 description: t("Adds two numbers together."),
 args: [
   arg("value1 (number)", t("The first number")),
   arg("value2 (number)", t("The second number")),
  returns: ["NUMBER"],
  compute: function (value1, value2) {
```

Declare a function

- Declare the arguments
- Implement compute
 - return a primitive value
 - return a 2d array of primitives
- Cast user inputs to required type
 - o toNumber("20%", locale) // 0.2
- Add the function to the registry
- References work out of the box

```
import { helpers, registries } from "@odoo/o-spreadsheet";
import { t } from "@web/core/110n/translation";
const { arg, toNumber } = helpers;
const MY ADD = {
  description: t("Adds two numbers together."),
 args: [
   arg("value1 (number)", t("The first number")),
   arg("value2 (number)", t("The second number")),
  returns: ["NUMBER"],
  compute: function (value1, value2) {
    value1 = toNumber(value1, this.locale)
    value2 = toNumber(value2, this.locale)
registries.functionRegistry. add("MY.ADD", MY ADD);
```

Our function



- Returns a 2D matrix of values
- Each column is a question and answers
- How to fetch the data?

```
import { t } from "@web/core/l10n/translation";
import { helpers, registries } from "@odoo/o-spreadsheet";
const { arg, toNumber } = helpers;
registries.functionRegistry. add("ODOO.SURVEY", {
  description: t("Return the results of a survey."),
  args: [arg("survey id (number)", t("The survey id"))],
  category: "Odoo",
  returns: ["RANGE"],
  compute: function (surveyId) {
    surveyId = toNumber(surveyId, this.locale);
});
```



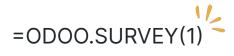
- Private state
- Introduce new getters to make parts of its state available
- Dispatch commands
- Some existing plugins:
 - O CellPlugin
 - ConditionalFormatPlugin
 - BordersPlugin

For our use case

- Acts as a proxy between the function and the data source
- Adds one getter getSurveyResults

```
import { registries, UIPlugin } from "@odoo/o-spreadsheet";
export class <u>SurveyPlugin</u> extends <u>UIPlugin</u> {
  static getters = ["getSurveyResults"];
  getSurveyResults (surveyId) {
registries.featurePluginRegistry. add (
  "SurveyPlugin",
  SurveyPlugin,
);
```

Our function



- How to fetch the data?
 - With our new plugin getter

getSurveyResults

```
/** @odoo-module **/
registries.functionRegistry.add("ODOO.SURVEY", {
    // ...
    compute: function (surveyId) {
        surveyId = toNumber(surveyId);
        return this.getters.getSurveyResults(surveyId);
    },
});
```



Fetch the data

- Function evaluation is synchronous
- Re-evaluate cells when data is fetched

```
import { registries, UIPlugin } from "@odoo/o-spreadsheet";
import { ServerData } from
"@spreadsheet/data sources/server data" ;
export class <u>SurveyPlugin</u> extends <u>UIPlugin</u> {
  static getters = ["getSurveyResults"];
registries.featurePluginRegistry. add ("SurveyPlugin",
SurveyPlugin);
```



Fetch the data

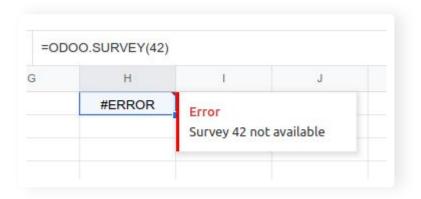
- Function evaluation is synchronous
- Re-evaluate cells when data is fetched
- Use ServerData
 - handles caching
 - notifies when data is fetched

```
import { registries, UIPlugin } from "@odoo/o-spreadsheet";
import { ServerData } from
"@spreadsheet/data sources/server data";
export class SurveyPlugin extends UIPlugin {
 static getters = ["getSurveyResults"];
  constructor(config) {
   super (config);
    const services = config.custom.env.services;
   this.serverData = new ServerData(services.orm, {
     whenDataIsFetched : () =>this.dispatch ("EVALUATE CELLS"),
   });
 getSurveyResults (surveyId) {
    const data = this.serverData.get(
     "survey.survey",
      "get survey results for spreadsheet" ,
   );
   return data;
```



Fetch the data

Error cells



```
export class SurveyPlugin extends UIPlugin {
 getSurveyResults(surveyId) {
    const data = this.serverData.get(
      "survey.survey",
      "get survey results for spreadsheet",
    );
    if (!data) {
      throw new Error(
        t("Survey %s not available", surveyId)
      );
    return data;
```



Return the data

- can be
 - a single primitive value
 - a 2d matrix of primitive values
- Normal access rights apply

```
from odoo import models,
class Survey (models.Model):
 def get survey results for spreadsheet(self):
    """Returns a 2d array of the results of the survey"""
   self.ensure one()
   if not self.search([("id", "=", self.id)]):
   result = []
   all answers = self.user input ids.user input line ids
   data = self.question ids. prepare statistics(all answers)
    for question results in data:
     question = question results["question"]
     answers = question results["answer line ids"]
        [question.title]
       + answers.mapped("display name"))
     result. append (column)
   return result
```



Performance pitfalls



Request at every evaluation

- You don't control when cells are evaluated
- Cache http request results
 - Managed by ServerData

One request per cell

Batch requests



Batch requests

```
this.serverData.batch.get(...);
```

Input

- list of ids (self)
- list of arbitrary args (using @api.model)

Output

- list of results
- Avoid Exceptions. The entire batch would fail

```
const data = this.serverData.get(
  "survey.survey",
  "get survey results for spreadsheet",
  [surveyId]
const data = this.serverData.batch.get(
  "survey.survey",
  "get survey results for spreadsheet",
 surveyId
);
def get survey results for spreadsheet(self):
    survey. get result table()
    for survey in self
def get result table(self):
```

Other APIs

=WORD.EXIST("hello")

Exact same principles

- Any HTTP request
- Cache results
- Batch requests (if you can)
- Don't forget security concerns

```
import { t } from "@web/core/110n/translation";
import { registries, UIPlugin } from "@odoo/o-spreadsheet";
export class DictionaryPlugin extends UIPlugin {
  static getters = ["doesWordExist"];
  cache = \{\};
  doesWordExist (word) {
    if (this.cache[word] === undefined) {
      this .fetch (word);
    switch (this.cache[word].status) {
      case "pending":
      case "rejected":
        throw this.cache[word].error;
      case "fulfilled":
        return this.cache[word].found;
registries.featurePluginRegistry. add (
  "DictionaryPlugin", <u>DictionaryPlugin</u>
);
```

Other APIs

=WORD.EXIST("hello")

Exact same principles

- Any HTTP request
- Cache results
- Batch requests (if you can)
- Don't forget security concerns

```
export class <u>DictionaryPlugin</u> extends <u>UIPlugin</u> +
  fetch (word) {
    this.cache[word] = { status: "pending" };
    fetch (`https://api.toys/api/check dictionary?text= ${word}
      .then((response) => response.json())
      .then ((data) => \{
        this.cache[word] = {
          status: "fulfilled",
          found: data.found,
      .catch((error) => {
        this.cache[word] = {
          status: "rejected",
          error,
      })
      .finally(() \Rightarrow {
        this.dispatch("EVALUATE CELLS");
```

Thank You!

Code available https://github.com/LucasLefevre/spreadsheet-functions-oxp2023

Documentation https://github.com/odoo/o-spreadsheet

