



# Explore Interaction Studio

Server-Side Code Tips and Tricks

**Mateusz Dąbrowski**

Marketing Cloud Architect & Manager at PwC  
Author at [mateuszdabrowski.pl](http://mateuszdabrowski.pl)





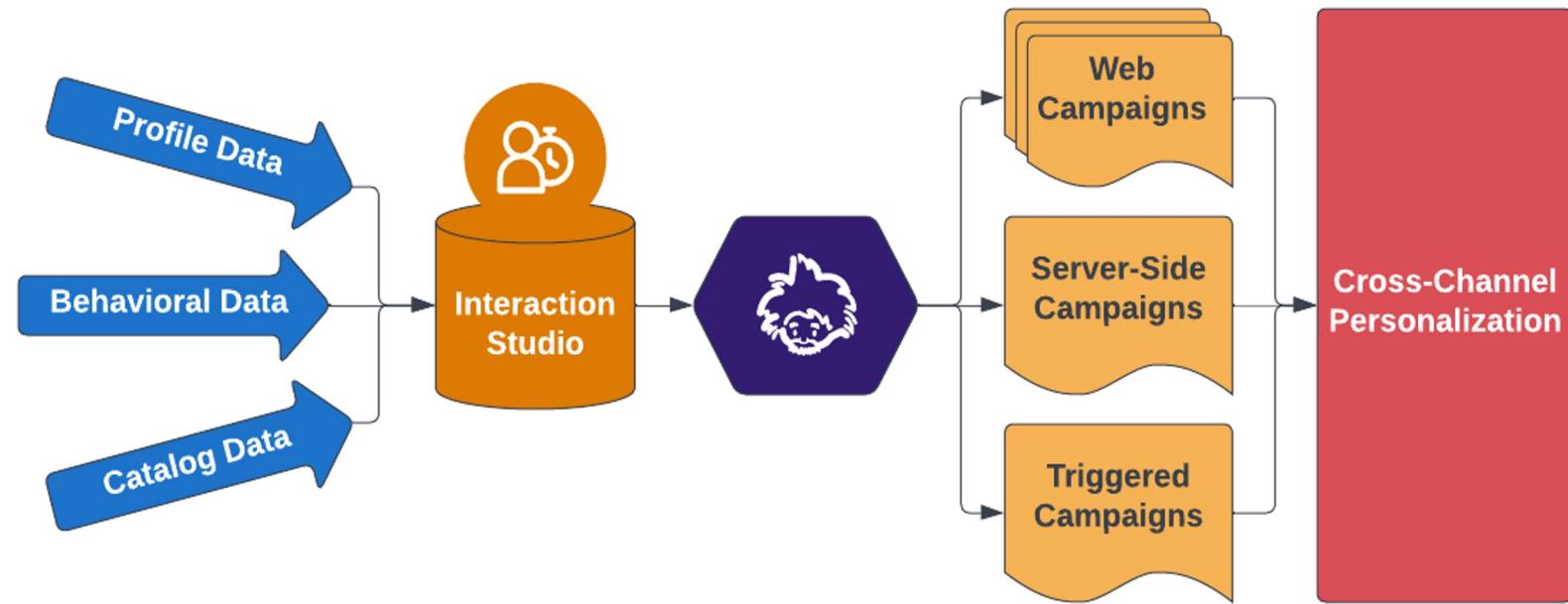
# Role of Interaction Studio

Salesforce Marketing Cloud  
Real-Time Personalisation



# Role of Interaction Studio

Marketing Cloud Real-Time Personalisation



# Role of Server-Side Code

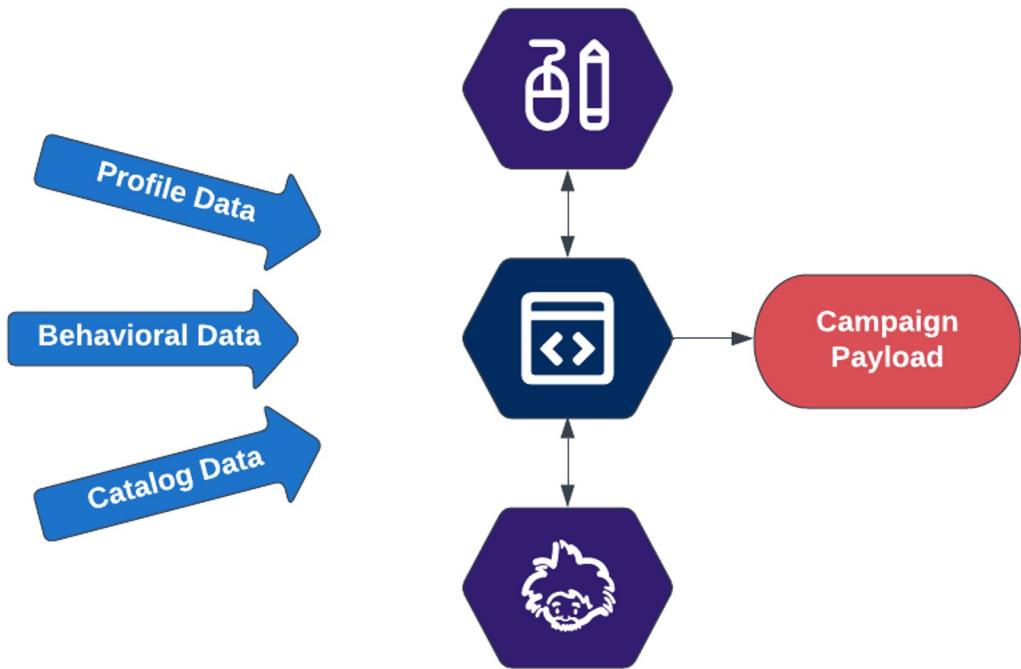
Hidden power of Interaction Studio



# Role of Server-Side Code



Hidden power of Interaction Studio





# Server-Side Tips & Tricks

Make the most out of Server-Side Code



# Server-Side 101

## Configure Campaign UI, Enrich your Data and Transform the Payload



New Template VIEWING DRAFT

**PUBLISH** **UNPUBLISH** **SAVE** **SAVE & CLOSE** **CLOSE**

**RENDER** **RESET**  Auto Render

Example Field  
Hello World

**TEMPLATE EDITOR**

Serverside Code

```
1  export class NewTemplate implements CampaignTemplateComponent {
2
3    @title("Example Field")
4    header: string;
5
6    run(context: CampaignComponentContext) {
7      return {};
8    }
9
10}
11
12
```

**Debugger**

Payload Preview Log Config JSON

Simulate

User  
Search for a User

Current Item Type  
Select an item type...

Current Item  
Select an item type first to then select an item

PAYOUT PREVIEW READ ONLY Documentation

```
{
  "contentZone" : null,
  "header" : "Hello World",
  "campaign" : "testCampaignId",
  "experience" : "testExperienceId",
  "userGroup" : "testUserGroup"
}
```

# Complex Properties



Group properties and capture related values.

```
export class ColorComplexProperty {  
  colorStartTime: DateTime;  
  colorValue: Color;  
}
```

```
colorConfig: ColorComplexProperty;
```

Color Config

Color Start Time

05/19/ Central European Summer Time

Color Value

#a32626

```
{  
  "colorConfig" : {  
    "colorStartTime" : {  
      "dateTime" : [  
        0 : "2023-05-19T10:00:00.000Z"  
      ]  
    }  
  },  
  "colorValue" : {  
    "hex" : "#a32626"  
    "r" : 163  
    "g" : 38  
    "b" : 38  
    "a" : 1  
  }  
}  
"contentZone" : NULL
```

# Arrays



Capture multiple inputs during campaign configuration.

```
colorConfig: ColorComplexProperty[];
```

Color Config

Color Start Time ×

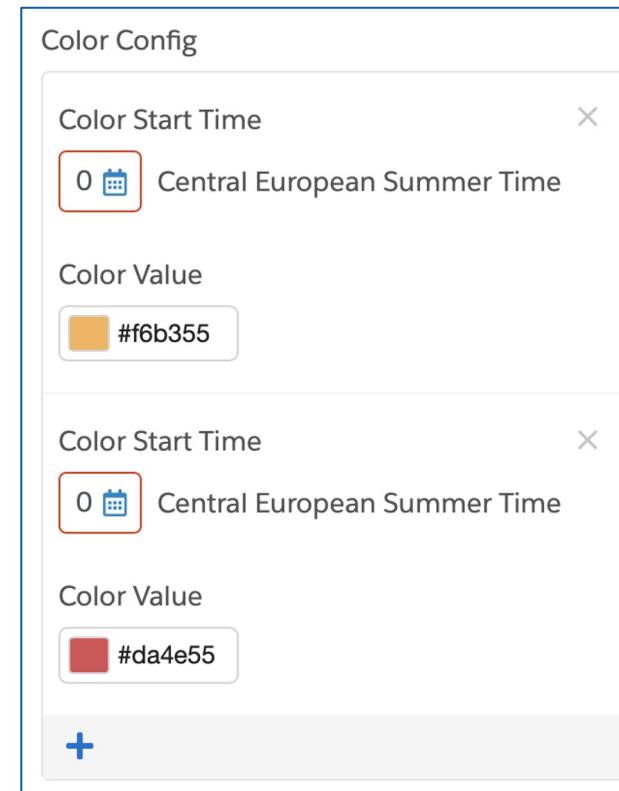
0 Central European Summer Time

Color Value #f6b355

Color Start Time ×

0 Central European Summer Time

Color Value #da4e55



```
{
  "colorConfig": [
    {
      "colorStartTime": { ... },
      "colorValue": { ... }
    },
    {
      "colorStartTime": { ... },
      "colorValue": { ... }
    }
  ],
  "contentZone": null
}
```

# Default Values



Set best practices and minimise the work.

```
@title("Example Field")
header: string = 'Ahoj';
```

Example Field

```
colorProperty: Color = {
    hex: '#F6B355',
    r: 246,
    b: 179,
    g: 85,
    a: 100
};
```

Color Property



```
▼ "colorConfig" : [
    ▼ 0 : {
        ▶ "colorStartTime" : {...}
        ▶ "colorValue" : {...}
    }
]
```

# Options



Control configuration and force branding / best practices.

```
logo: 'horizontal white'  
| 'horizontal black'  
| 'vertical white'  
| 'vertical black';
```

Logo

Select...  
horizontal white  
horizontal black  
vertical white  
vertical black

```
export class ComplexLogo {  
  orientation: 'horizontal' | 'vertical';  
  color: 'white' | 'black';  
}
```

```
@options([  
  'horizontal white',  
  'horizontal black',  
  'vertical white',  
  'vertical black'  
)  
logo: String;
```

"logo" : "horizontal white"

```
Complex Logo  
Horizontal White ▾  
  Horizontal White  
  Horizontal Black  
  Vertical White  
  Vertical Black
```

```
  "complexLogo" : {  
    "orientation" : "horizontal"  
    "color" : "white"  
  }
```

# Decorators



Control descriptions, UI structure and input visibility.

```
@title('Welcoming message')
@subtitle('Make it yours')
sayHello: string = 'Ahoj';
```

Welcoming message

Make it yours

```
@range(true)
dateProperty: DateTime;
```

Date Property

05/17/: Central European Summer Time

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10
12	:	00	PM			

```
export class Configuration {
    @title('Show Overlay after Delay')
    isOverlayDelayed: boolean = false;

    @shownIf(this, (self) => self.isOverlayDelayed)
    @optional(true)
    @units("seconds")
    campaignDelay: number = 3;

    @optional(true)
    @title('Show Overlay on Exit Intent')
    isOverlayOnExitIntent: boolean = false;
}
```

Pop Up Configuration

 Show Overlay after Delay  
 Show Overlay on Exit Intent

Pop Up Configuration

 Show Overlay after Delay  
Campaign Delay  
 seconds  
 Show Overlay on Exit Intent

# Recommendations



Leverage Recipes to their fullest potential.

```
import { RecommendationsConfig, recommend } from "recs";
```

```
recommendationConfig: RecommendationsConfig = new RecommendationsConfig();
```

Recommendation

Item Type

Product

Recipe

Recipe - Collab Filtering

Max Results

12

```
let recommendations: any = recommend(context, this.recommendationConfig);
```

```
@title('Main Recommendation')
recommendationConfig: RecommendationsConfig = new RecommendationsConfig()
  .restrictItemType("Product")
  .restrictMaxResults(6)
```

Main Recommendation

Recipe

Recipe - PDP Product Recs

```
@shownIf(this, (self) => self.AddAdditionalRecommendationConfig)
@title('Optional Additional Recommendation')
additionalRecommendationConfig: RecommendationsConfig = new RecommendationsConfig()
  .restrictItemType("Product")
  .restrictMaxResults(6)
```

```
let additionalRecommendations = recommend(context, this.additionalRecommendationConfig);

recommendations = recommendations.concat(
  additionalRecommendations
    .filter(additionalProduct => recommendations
      .map(product => product.id)
      .indexOf(additionalProduct.id) < 0)
)
```

# Context

Enrich your payload with real-time data.



```
run(context: CampaignComponentContext) {  
  return {};  
}
```

```
  ↴ accountId (property) GearContext...  
  ↴ beaconVersion  
  ↴ campaignId  
  ↴ configuration  
  ↴ contentZone  
  ↴ datasetId  
  ↴ event  
  ↴ experienceId  
  ↴ locale  
  ↴ services  
  ↴ trigger  
  ↴ user  
  
context.
```

```
if (context.contentZone === 'homepage_hero') {  
  
  if (context.trigger?.type) {  
  
    let cart = context?.user?.currentCart;
```

```
let isCartAbandoned: Boolean =  
  cart?.updated.getTime() < Date.now() - 21600000  
  && cart?.updated.getTime() > Date.now() - 60480000;  
  
if (isCartAbandoned) {  
  let cartItems: string[] = context.services.catalog  
    .findItems('Product', cart.lineItems.map(cartItem => cartItem.itemId))  
    .filter(cartItem => cartItem?.attributes?.inventoryCount?.value != 0)  
    .map(cartItem => cartItem.id);
```

christopher-  
longSFDC/mcp



Helpful examples for Salesforce Marketing Cloud  
Personalization.



# Imports



Unlock hidden functions of Server-Side Code.

```
import { RecommendationsConfig, RecipeReferenceLookup, RecipeReference, recommend } from 'recs';
import { UserAttributeLookup, UserAttributeReference, UserSegmentLookup, UserSegmentReference } from 'common';
```

```
@title('User Attributes')
@lookupOptions(() => new UserAttributeLookup())
attributes: UserAttributeReference[] = [
  {
    "id": "emailAddress",
    "label": "emailAddress (Email Address)",
    "type": "String"
  },
];
```

```
@title(' ')
@shownIf(this, (self) => self.includeSegments)
@lookupOptions(() => new UserSegmentLookup())
segments: UserSegmentReference[];
```

Include User Segments

Select only few critical segments.

Select...

No Visit for 14 Days

Low Engagement Score (<25%)

User Attributes

emailAddress (Email Address) x loy ▾

loyaltyID (Loyalty ID)

loyaltyTier (Loyalty Tier)

```
static getUserSegments(segments: UserSegmentReference[], user: User) {
  if (segments?.length && user?.segmentMembership?.length) {
    const selectedSegments: string[] = segments.map((segment) => segment.id);
    return user.segmentMembership
      .filter((memberSegment) => selectedSegments.includes(memberSegment.segmentId))
      .map((memberSegment) => memberSegment.segmentName).join(',');
  }
  return '';
}
```



Thank  
you