



ColorCloud

HAMBURG

24. - 25. APRIL 2025

Bye Boilerplate:
Entwicklung von
REST API Clients für
Business Central
automatisieren

Simon Fischer

Tobias Fenster

ColorCloud Sponsors

THANK YOU! 

Unicorn Sponsor



Gold Sponsors



Silver Sponsors



Digital Sponsors



Special Sponsors

Learn4D365®





Simon Fischer

Senior Software Developer, 4PS Germany

 Hamburg, Germany

 SimonOfHH

 [simonofhh.bsky.social](https://bsky.app/profile/simonofhh.bsky.social)

 sfischer@4ps.de

 [SimonOfHH.tech](https://simonofhh.tech)

#ColorCloudSpeaker 





Tobias Fenster

GF 4PS Deutschland, Chief Engineer Hilti

Microsoft MVP & RD, Docker Captain

Blog und Podcast tobiasfenster.io

LinkedIn und Bluesky



BY **HILTI**

#ColorCloudSpeaker 

Agenda

- Einführung und typischer Verwendung Kiota
- Kiota-Clients mit BC / AL
- Helft uns!

Die Herausforderung: API Nr. 3928

- Mittlerweile (zum Glück) viele Integration auf API-Basis
- Handling in AL verbessert
- Dennoch: sehr viel repetitiver Code notwendig, der keine Business-Logik trägt
- Sich ändernde, erweiterte APIs benötigen manuelle Wartung

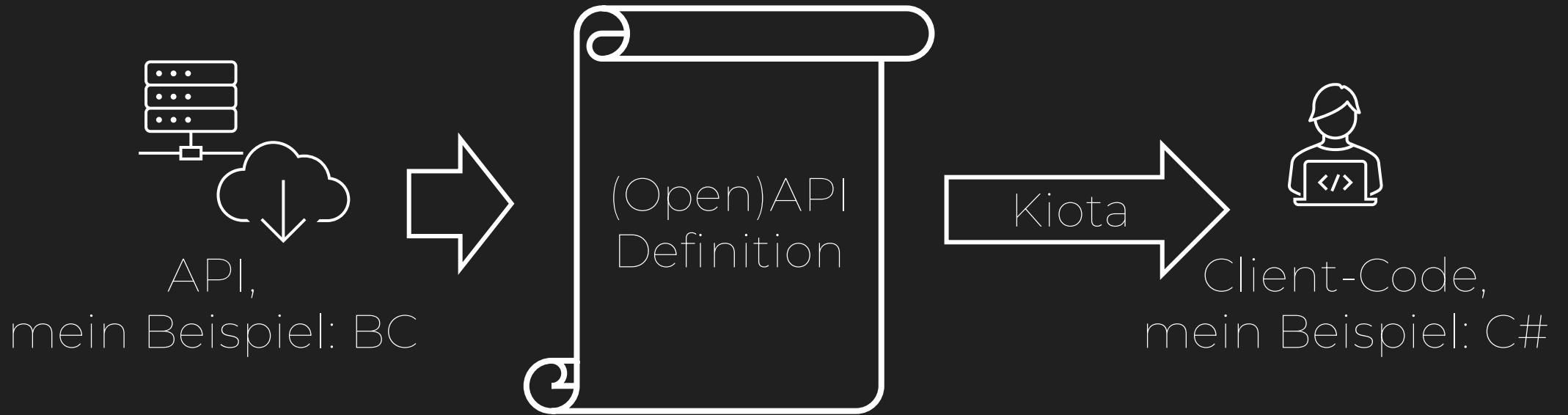


Eine Lösung: Kiota generiert Clients

- Statt manuell geschriebenen und gewartetem Code: Clients mit Kiota generieren
- Idee: OpenAPI-Definition einer API verwenden, daraus Client generieren
- Nicht vollständig notwendig, kann auch nur Teil der Definition sein
- Generierung in unterschiedliche Programmiersprachen möglich



Eine Lösung: Kiota generiert Clients



Demo time!

C#-Client für Business
Central API



Wie hat das funktioniert?

1. OpenAPI-Definition generieren wie [in meinem Blog](#) beschrieben, basierend auf einem [Blog Post von Waldo](#)

```
New-BcContainer -accept_outdated -accept_eula  
-containerName demo-colorcloud -imageName mybc  
-artifactUrl (Get-BCArtifactUrl -type Sandbox -country DE)  
-auth NavUserPassword -multitenant:$false -updateHosts  
-myscripts  
@"https://raw.githubusercontent.com/tfenster/nav-docker-  
samples/swaggerui/AdditionalSetup.ps1")
```

2. Entstandene Datei kopieren



Wie hat das funktioniert?

3. Kiota installieren

```
dotnet tool install --global Microsoft.OpenApi.Kiota
```

4. Client generieren über Kiota VS Code Extension

5. Client nutzen

Demo time!

Client generieren und
nutzen



Business Central



ColorCloud

www.colorcloud.rocks

#ColorCloud2025

Status Quo BC-API-Clients

- Typischerweise nicht sehr komfortabel zu nutzen (fehleranfälliges Arbeiten mit JSON-Objekten oder temporären Tabellen)

```
JToken.ReadFrom(Data);
JObject := JToken.AsObject();
JObject.SelectToken('results', JToken);
JArray := JToken.AsArray();
```

```
foreach JToken in JArray do begin
    NYTBestSellersTheme.Init();
    NYTBestSellersTheme."List Name" := CopyStr(GetValueAsText(JToken, 'list_name'), 1, MaxStrLen
    NYTBestSellersTheme."List Name Encoded" := CopyStr(GetValueAsText(JToken, 'list_name_encoded
    NYTBestSellersTheme.Updated := CopyStr(GetValueAsText(JToken, 'updated'), 1, MaxStrLen(NYTBe
    NYTBestSellersTheme."Newest Published Date" := DT2Date(EvaluateUTCDateTime(GetValueAsText(J
    NYTBestSellersTheme."Oldest Published Date" := DT2Date(EvaluateUTCDateTime(GetValueAsText(J
    NYTBestSellersTheme.Insert();
end;
end;
```

```
JsonOrder.Get('Lines', JsonTokenLines);
JsonArrayLines := JsonTokenLines.AsArray();
```

```
foreach JsonTokenLine in JsonArrayLines do begin
    Message(format(JsonTokenLine));
end;
```



Status Quo BC-API-Clients

- Je nach Umfang der API sehr viel repetitive Arbeit um Datenstrukturen abzubilden

```
// To store the content in JSON Object  
JObject.ReadFrom(Result);  
JObject.Get('name', Jtoken);  
API.Name := Jtoken.AsValue().AsText();  
JObject.Get('email', Jtoken);  
API.Email := Jtoken.AsValue().AsText();  
JObject.Get('username', Jtoken);  
API."User Name" := Jtoken.AsValue().AsText();  
JObject.Get('street', Jtoken);  
API.Street := Jtoken.AsValue().AsText();
```



Status Quo BC-API-Clients

- Bestimmte Schritte (Authentifizierung, Http Header-Handling, Content-Aufbau, ...) werden zum Teil immer noch jedes Mal fehleranfällig neu implementiert (besser seit AJs Rest Client Module)

```
HttpRequestMessage.Method := 'POST';
HttpRequestMessage.SetRequestUri("ptsv2.com/.../post");

HttpContent.GetHeaders(HttpHeadersContent);
HttpHeadersContent.Clear();
HttpHeadersContent.Add('Content-Type', 'application/json');

HttpRequestMessage.GetHeaders(HttpHeadersRequestMessage);
HttpHeadersRequestMessage.Add('Cache-Control', 'no-cache');
HttpHeadersRequestMessage.Add('Authorization', 'bearer ' + Token);

JsonObject.Add('Kind', 1);
JsonObject.Add('ActorCSID', 74);
JsonObject.Add('ProdConceptID', 38);
```



In a galaxy not too far away...

- Wäre es nicht viel schöner das JSON nicht manuell zu parsen...

```
procedure ReadItemNumberSchemaIdentifierFromProjectJson(jsonBody: JsonToken): Text
var
  subTokenProject, subTokenProjectInfo, subTokenItemNumberSchema, subTokenIdentifier : JsonToken;
begin
  if jsonBody.SelectToken('project', subTokenProject) then
    if subTokenProject.SelectToken('projectInformation', subTokenProjectInfo) then
      if subTokenProjectInfo.SelectToken('itemNumberSchema', subTokenItemNumberSchema) then
        if subTokenItemNumberSchema.SelectToken('identifier', subTokenIdentifier) then
          exit(subTokenIdentifier.AsValue().AsText());
        exit('');
      end;
    end;
  end;
```

- ... sondern einfach einen direkten Zugriff zu haben ...

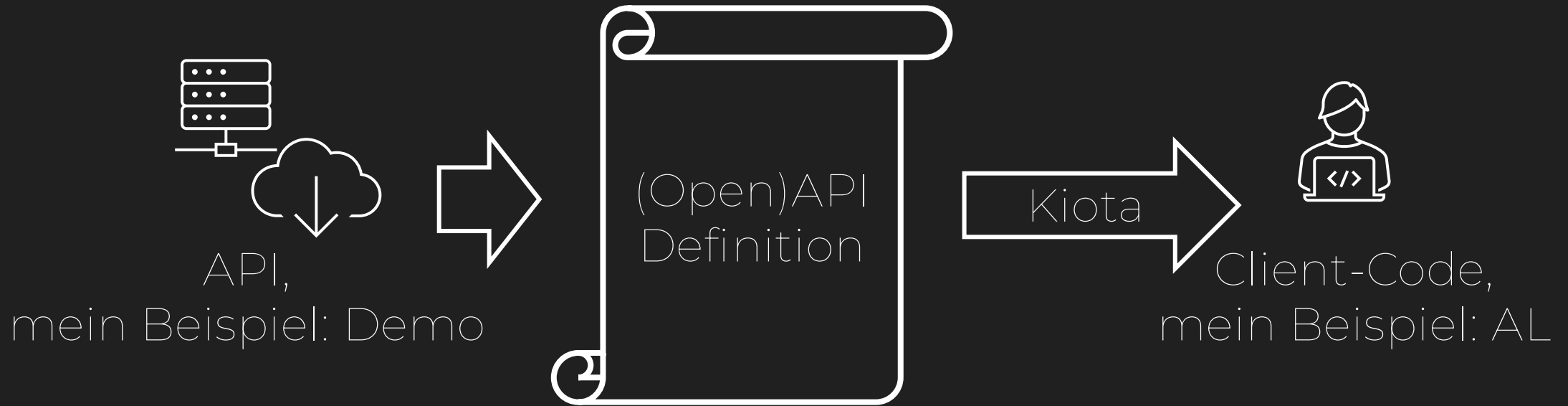
```
procedure ReadItemNumberSchemaIdentifierFromProject(Project: Codeunit "ProjectDto"): Text
begin
  exit(Project.ProjectInformation().ItemNumberSchema().Identifier());
end;
```

In a galaxy not too far away...

- ... Zeit für die Business Logik zu haben, statt Objekte aus der swagger.json manuell abzubilden ...
- ... wertvolle Entwickler-Zeit nicht für mind-numbing Tasks zu verschwenden ...
- ... endlich Tooling zu nutzen, was in anderen Frameworks/Sprachen nicht wegzudenken ist?



Eine Lösung: Kiota generiert Clients



Demo time!

Kiota für Business Central



ColorCloud

www.colorcloud.rocks

#ColorCloud2025

Wie ist der aktuelle Stand?

- Es gibt einen Fork des microsoft/kiota Repositories ([GitHub](#))
- der aktuelle Stand (und die Abweichungen zum üblichen Kiota-Processing) sind hier beschrieben (auch ein hoffentlich einfacherer Einstieg in die riesige Codebase): [README.md](#)
- Es gibt eine „Companion“-Extension für BC um so wenig wie möglich duplicated code zu haben ([GitHub](#))



Wie ist der aktuelle Stand?

- Jetzt brauchen wir euch - die Community!

[\(PR\)](#)

- Feedback - sieht der aktuelle Stand brauchbar aus?

- Stabilität - wer testet mit uns?

- Make some noise - wie bekommen wir das am schnellsten bei Microsoft gemergt (sowohl die Kiota-Änderungen als auch die BC-App)? Indem MS sieht, dass die Nachfrage da ist





ColorCloud

HAMBURG

24. - 25. APRIL 2025

THANK YOU!



ColorCloud

www.colorcloud.rocks

[#ColorCloud2025](https://twitter.com/ColorCloud2025)



ColorCloud

HAMBURG

24. - 25. APRIL 2025

Q & A



ColorCloud

www.colorcloud.rocks

[#ColorCloud2025](https://twitter.com/ColorCloud2025)



ColorCloud

Feedback

Please rate our session to help us improve!

Take part in ColorCloud Raffle 🔥
Every submitted feedback counts as one raffle ticket! Good luck 🍀



ColorCloud

www.colorcloud.rocks

#ColorCloud2025