



Speaker 1

Today is session number nine of our development on Creature platform guided learning. And today we will continue to study server side and we'll move on with integration tools. So yesterday we finished with our own web service. It was made at creation site, was made with the help of C Sharp sources and we programmed it, we saved it in file system. We used Visual Studio to develop this web service. So we used our examples. Now you know, okay, now you know how you can make your own integration tools. But in general, so integrations will require much more entities and will require much more tasks to exchange data. So it will be really hard for you to write a web service for each data transfer that you need in your system.



Speaker 1

So what we'll study today, how to use standard platform level tools to operate with data with the help of HTTP queries. And we will discuss and I will show you examples of how to work with standard tools. And they are all data protocol and data web service. Also I plan to show you how to call third party web services with the help of no code tools, with the help of settings for web services and call web service item. And at the end of the session I want to tell you about Clio tool, Clio Commons. You will understand what is this, how it can be used, why you need some additional tools. So we will discuss a bit more about system maintenance delivery and if you have any questions, I'll be really happy to hear and to answer them.



Speaker 1

So don't be shy, ask any questions if you feel something that you need to know and let's move on. So integrations with data tools. First of all I need to show you general integration capabilities that we have on board. Go to development guides integrations options and here you can see standard options that we have. We already studied custom web service option and it offers us possibility to program anything we need to ask for data, to make operations with files, to use any libraries. But this approach obviously requires some programming at creation site. And also this approach requires programming at third party site in order to normally correctly call such service and to get response data parse such response. So I just want to say that in general it's quite expensive approach if you need to do a lot of different operations.



Speaker 1

As you can see we have other options to integrate with creation. We have two options for data transfer such called CRUD operations which means create, read, update and delete. So standard simple operations that you may need to work with your data and to organize. Possibility to read some data for third party app or to make some inserts, updates or deletes in creation by commands from other applications. And we have two different options for this. One is called Data Service, another is called Odata. And Odata is quite common because it was designed by Microsoft and it's possible to find Odata clients as third party sources. There are a lot of tools that understand how to work without data. So developers of Creation decided to support this type of client. And data service is much more unique.



Speaker 1

It's very specific to creature, but it offers you more options, more possibilities, more complicated calculations. And that's why data services used for creation of client side and data service in general is more powerful than ODATA and more complex to program. So I will show you both ODATA and Data Service. You will see how it works. Also, it's worth it to mention that we have another integration option to run business processes. So you can use process engine service web service for starting or continue execution of business processes the same way how you can do it from creation client side. Physically, this web service is the only one to handle processes. And when we run processes from inner application. Let me remind you how we did it.



Speaker 1

We had the class on Freedom UI section, we can open its page and then we made an action to calculate average price with the help of business process. You remember we did this calculation, we started process, now we have some results, average price in dollars. Then we save it or close it. And that's how we finish our process. So physically we did some calls to third part to process engine web service. Here you can see it. And the same calls with the same parameters can be done with third party applications. So it's possible to run processes from 30 third party. And of course in this case you need to pass authentication first. And we did it yesterday with the help of Postman tool. You remember we had the Postman authentication, we had set of authentication cookies. Okay.



Speaker 1

So if interested, you can just watch yesterday video and get more details. So today we'll focus on data transfer integration options. And I would like to start from OData because it's easier, it's more friendly for beginners and it's quite easy to start from scratch. So what it is, this is a data transfer protocol which is supported at creation. We have all documentation about it. So for all data we have documentation explaining how to use it. And in general you need to know that creation supports OData 4 and OData 3. Unfortunately, OData 3 is not supported in packages. So it's not supported for objects that are saved in packages compiled as Separate assembly. So it means that for your ODATA integration, probably it's better to focus on OData 4 from the beginning.



Speaker 1

We have a lot of documentation about it and if interesting you can find much more. And we have examples, we have references. So I prefer to show you something really useful. This article, it explains some general basics about how to use OData. We will try to make some examples and first you need to know that ODATA operates with the help of Data Model. So it means it respects all existing objects, their columns, their names and so on. So when you operate with OData, as well as when you operate with Data service, it uses Data Model. So it uses information about existing objects, columns, references and access rights restrictions.



Speaker 1

Okay, so depending on what operation you need, you have to select the proper HTTP method, Get is used to select post for inserts, patch for updates, and HTTP Delete is used to perform physical delete operation. I will show you a couple of examples and also you need to know that we have a lot of interesting documentation samples here. And one of the good ones is creation API documentation hosted at Documentor, get postman. Com. This is one of the best sources that you can find for OData. I plan to show you examples with OData 4. As you can see, first of all, we have to make authentication correctly. Then we can do different queries and different examples. So depending on your task, you may find a corresponding sample here. And also we have a lot of samples to make different filters.



Speaker 1

You see, for different filters there are special expressions in let's say ODATA language. And so you can find a lot of examples here. Even batch queries Supported Batch queue means running single query with several parameters. And each parameter explains to a system how to run a particular operation. So it can be useful if your plan is to run many data operations with one single query. So batch operations are also supported. But we can do it. Let's not go too far. And I understand that probably not all of you will start your integration at all. And this is something that you just need to know and let's say get familiar with it. You will see how it works. So let me show you examples. My plan is to make some selection of data. We have some examples of data records.



Speaker 1

Let's read data from our reality object, but with the help of all data. First of all, I will use integration tool, I will use

Postman app and my yesterday cookies are not working anymore. They will not help me to run queries. I can check it easily so we can Try to run our web service and as you can see we have 401 not authorized. It means that our cookies are related to expired session. So providing such cookies we will not be able to use any business methods any business logic methods of creation web services. So we have to get new cookies. I prefer to clear previously set cookies. We go to perform login operation again. Yesterday we discussed how to do this. We use special URL to all service and its login method inside of a root part of our application.



Speaker 1

We provide login and password. No any special data, no any special headers. You may find some information here like authentication and some accept content type for C session. But it turns out to be not necessary to perform authentication successfully. So let me show you here we have some hidden headers made by postman like accept content type, application JSON and so on. But so I prefer to keep it as is by default. And we have login and password. Originally we have no cookies. So let's try to get new session for us. Yes, we got it. 200 no errors. So currently we have set of cookies enough for us to perform inner queries. Okay, we can create new query and our task. Let's keep it easy and keep it simple. So we will try to read data.



Speaker 1

We will have to make a selection of data records and we will make it get HTTP query. Let's do this. So let's go to postman, create a new get query. We have to use our address of our app with 0 alias here and then we can look at this example or we can look at this document or get postman.com examples. So we need to do selection. And our part for odata web service is odata here. So it's a name of odata endpoint then/in odata4 we should use just entity name here. It's not obvious that collection one is entity name. And here you can see that it's also not so obvious, but maybe you can find examples. Yeah, here you can see an example. So it will make it clear how we can do this. So I will use an example with my USR realty.



Speaker 1

If I provide no any parameters, system will try to read all of the data with all of the records, all of the columns. Let's try to do this and run send for get queries we do not need to use BPM CSRF protection. So for get queries, no cross site request forgery protection needed. And you see it took significant time, almost six seconds to run. And as a result we have Some JSON body with different data records about our reality record saved in database. As you can see all of the columns, a lot of unnecessary data. But that's how it works. If we provide no special parameters. Now I will show you how we can make it more interesting and more useful. We can use different parameters in our script in our query.



Speaker 1

And let me show you for example parameters for selecting data collection instance selected fields. So here I can see an example and it means dollar select so question dollar select and then we have columns with comma separated values to get necessary columns. So I'm using this example, you can use this example and let me show you how we can do it. So question mark to switch to parameters Dover select is parameters specifying column names that we plan to use. Don't forget we have to use column codes, not titles USR name USR price USD maybe created on. So if you want you can get more. It's possible to use lookups in order to get corresponding names of type or name of a person who created the record. But now I try to keep it simple. So this is example of how we can run queries.



Speaker 1

You see only requested columns. Now here and we can see all of the data. We can do some limitation. So we can use and another parameter dollar top three for example. And when we do such selection, it will read only top three records for us. You see on the top three records you can use sort, you can use filter. We can use x order by in order to organize some sorting. We can use special parameters to select only one specific record but filtered by ID a lot of other options. So if necessary you will be able to do this. And I try just to keep it simple and quite useful for you. So

here is an example of a get query to read some data. So you can try to do it even in your browser.



Speaker 1

In case if you will try to run such query in your browser, you do not need authentication because usually your browser is already authenticated in creation and you keep running session running pages of your creation in your browser. And that's why you don't usually need to prepare separate authentication for it. Okay, I think for selection it's quite clear. So let me show you how you can do insert data. I need to use almost the same set of parameters but for insert. As you can see here we have post to add data. Let's do this. We can also look at examples for post and our URL will be quite simple. We will just provide post and this URL which includes our website zero application Alice OData is the endpoint for OData Web Service.



Speaker 1

Then user really is also part of endpoint and it gives information to data what exact object you plan to operate with. Then I have to look at body row JSON and we have to provide some data for example user name. This is mandatory data from all data for integration it will be our name user price USD okay, so you can see we have some data here. If we try to run a post query without CSRF settings we will get 403 because for post query we must perform BPM CSRF header we must put corresponding cookie value. Please carefully copy cookie value including some dots if they present. So it's important in this case our insert will work. Okay? No, it tells that comment doesn't exist. Okay, it was my fault because common name is user comment. Okay, everything is working.



Speaker 1

Now we can go to our main app, sort by date of creation and we will immediately see that our data was added and we can also check column values. Here you see price was added, we see column was added correctly. But what we really interested you can see the type and offer type were set by default. It means our object model worked well for us and moreover our object model also supported all event handling provided with help of start signals. Now you see three realty visits were already planned for upcoming days. So all business logic that was designed and programmed with the help of event handling with the help of start signals it will work for ODATA as well and server side handling like we did to validation for very big prices will also work. Now let me show you how it can look.



Speaker 1

So let's try to add very big number and it will be more than 1 billion. So we will see how system will react on this attempt. Now you see we have a quite rude error and this is error 500 internal error. But we have error text which means we intentionally erased such exception. We provided error message text of what is wrong and of course physical insert was not done so we can read data, you see no new records created. So this is example how you can use old data for your tasks and it will be really attractive for relatively simple tasks for your future integration needs where you need to transport not so many data records. If your task is to transport millions of rows, possibly such approach with running separate queries for each record will be not very effective, not very efficient.



Speaker 1

In case if your task is to transport huge amounts of data, you have the only one option. This is your own programming of such data and your own parsing your data structures and your own database direct operations and avoiding use of object model. Because when you use object model, you also support all of the object events. It works not so fast. And for example, when we did insert with the help of postman, it took almost 180. No, in total it took a bit longer. You see, we have some additional cost for preparation of the query. So it took 300 milliseconds. And it's important because if your task is to transport like 1 million of data rows, if you spend 300 milliseconds for each one, it means you will wait for several days and this speed probably will be not suitable for your customers.

Speaker 1



So you have to search for a better solution. So in case of big number of records to process, in case of very, let's say, hard speed requirements for your solution, the only suitable option is your own web service with your own data formats, your own direct database management without use of object model. But in simple real life scenarios, I think all data will work well for you. And it's relatively easy because you see it's not so hard to build such query to run it even when your third party app is used. So you can do such programming. And also you will get response data in JSON string, so you can easily parse it, you can easily extract corresponding results that you may need. Here you can see JSON objects and you can also get corresponding values from here. So it's very suitable for beginners.



Speaker 1

And for some tasks I think it will work perfectly. But for some other tasks you may find that all data is not enough, maybe because of hard load conditions you will finally realize that our data is not going perfect with hard, I would say hardwood big amounts of data. Maybe you will need some more complicated scenarios with reading data and performing some calculations like preparing reports and running it with third party app. And you will need some aggregation, some complex filtering. In this case, all data probably will be not the perfect solution. And data service may work better. For data service is created by creature developers, not by Microsoft. And this is unique to creature tool.



Speaker 1

So it's appropriate way of operating with data using creature server side, it can do all the same operations like create, read, update and delete data and but it can do it a bit more efficiently. And also it supports different filter condition macroses and it's much more rich when you need to select data from different tables, perform some aggregations, perform some complex filtering. I will demonstrate what data service can do. And first of all I need to tell you the data service is mainly Used by inner creation client side pages. Each time when you operate with any list or edit page, you will see data service web service queries. You will also see that payload for that queries. It means arguments that you need to transport. Payload is quite complex. You see more than hundred of different settings and if you preview.



Speaker 1

So if you view source, you see a lot of settings and parameters that can be passed as arguments for data service. But it also has a response with a JSON. So now you see JSON string which can be parsed and such JSON data not so hard to operate with. You can easily get corresponding values out from your results. So Data service web service maybe can be suitable in case if for some reasons old data doesn't work for you as you expect. I will show you examples of data service but first I wanted to mention that it's used inside of creatio pages. So data service was designed for inner creation data operations with client side. All lists and all edit pages are using data service to get or to modify, insert, update any data record.



Speaker 1

So in documentation you will find that data service is promoted as integration tool. So it's a restful service. You can use third party to compose queries for it. And here it's promoted and advertised like an integration tool. But in reality it's not so integration oriented, but it is say platform UI oriented. Because all UI pages are working with the help of this web service. I will show you some advanced features that this web service supports and let me hide some unnecessary columns. So my plan is to show you special types of columns called aggregate columns and also a bit later aggregate filters. So aggregate column can make some calculation over connected data records. And such connections will be done with the help of main record. Let me show you what I'm talking about. So each reality record may have number of visits inside.

Speaker 1



Recently created records are having more visits because we already had automatic creation for them. But also we have. Oh, we don't make any action to add the data director here. Okay, so we have some examples which already have created data records in visits detail. And for each reality record we have number of visits, connected detail records. So we can make a career column. We can make an aggregate column which will simply calculate number of connected records. This is one of the easiest possible aggregate columns that we can make. So let me show you how you can do this.



Speaker 1

Add the columns and instead of working with traditional set of columns we have in main object you can go to related objects and then we will see all objects that are available for us according to lookup columns from main object and also we will see reverse joins. We will see other entities that have lookup columns pointing to reality. In my case this is reality visit. We have different options how to select data from it. And in last versions we have also advanced examples like reading top one record. For example, we can read top one comment sorted by date of creation. Or we can see top one customer who is specified in recent visits and so on. In my case I try to keep it simple. Let's look only at number of records of connected visits and click select.



Speaker 1

It's also possible to specify special conditions to select data. For example, we can only calculate number of records that are in future or only visits where contact is specified. So you can do different conditions here. In my case I also try to make it easy and calculate total number of visits. And we can make the caption here visits count. So this visits count will be our title for new created aggregate column. Then we click save and now you see new column which shows us some data. And for each separate reality we have some calculation. It's interesting to mention that this type of calculation is also performed with the help of data service. Moreover, such calculation is performed with the same query where a main data record is selected.



Speaker 1

So if you reload the data using this update refresh button, you see only one query was executed and if you go deeper in payload you will see that our column for calculation of aggregate number is also represented to the standard column. Here we have some column path for it, some setting for Type of aggregation. Aggregation Type 1 means count and that's how system knows what to calculate and returns as some calculated number. This it was example of aggregate column and it can be really useful. And you should understand that this selection is not performed from reality. This column is obtained as a result of subquery from reality visits. And I want you to see one more feature called folders and here we can make additional folders select and add the new folder here.



Speaker 1

So in general folders work here like search folders in your outlook. So here we can specify some name and the filter conditions that will be useful for us to select only some searching data, not all of the data from our list. Let's call it three plus visits. So let's imagine for some reasons we need to look at realty records where we have three and more visits created. We are not interested in realty records with no visits. So we can make such a name here, save it. Then we can provide filter conditions for this folder. So now I'm planning to show you so called aggregate filters. We already saw aggregate columns and you understand that they represent result of subqueries but calculated with the same main query.



Speaker 1

And aggregate filters will be used for selecting data applying conditions on connected records, not on reality data but on connected records. In my case I plan to select only reality data where this visits count is greater or equal to



three so we can make condition and here we have to select connected entity. So not just dropping down here contents of reality columns. We have to click on this plus in order to select connected entity. In my case connected entity is real and the aggregation type will be just quantity and also we have alternative options like maximum or minimum date for creation or modification. If we had integer or decimal values there then it's possible to calculate average price maximum or minimum.



Speaker 1

So in my case I will just do quantity calculation select and we have condition count greater or equal three that will be our filter condition. We also can make additional filters here like counting only visits in future or counting only visits with not empty comment. So whatever you think it will be useful you can do here. Then we save it and that's how we see result of this filtering. So all data shows us all reality records. This search folder shows us only records which follow corresponding filter conditions. Such filters are called aggregate filters and we can now go again to network and reward the data to see that our payload also has special filter condition. And this filter condition includes information that we use aggregate function.



Speaker 1

We use count for our aggregation so it will do selection only of records where some sub select some count of connected data is greater than certain right expression and we have just value three here. So my example is to demonstrate that data service web service is capable not only to read plain data from the data sections, but also it's capable able to calculate aggregate columns and to use aggregate filters and you can use data service for your integration. But I think it's really hard because you will need to have a good make already good made examples to make it work. So if you make corresponding queries in your browser then you can steal all necessary parameters. Let me show you how we can do this. We can copy request URL from our query that we spotted in our network tab.



Speaker 1

Then we go to postman create new actually you should be careful because data service is usually working with post queries only. So we go to postman. Regardless of the operation we will use post query and paste this URL should be careful. Okay Paste URL we have a huge body. We have go to payload view source select everything copy then we go to body here this is row JSON and we paste this big body it's really hard to analyze it so we can use beautify tool to see it in a more structured way. So now you see we have a lot of options here that are used by data service and required here to be present. And as you can see I did not type them manually. I only used existing example from my browser console.



Speaker 1

This is a post clear so we obviously will go will fail into CSRF protection if we do not care about it. So BPMCSRF header and corresponding cookie value you should get it carefully copying full value and pasting it here. Now we will successfully run our query, we have some valid response JSON body and we see set of records, we see set of data and in general if you run something like this from third party app then you'll be able to parse your data and you'll be able to analyze it and get corresponding numbers or other columns if you need it. Adrian is asking is folder the only way to go to Advanced filter in Freedom ui? Yes, Currently we don't have any separate advanced filters for data selection, so developer decided to keep it saved into folders.



Speaker 1

In Classic UI we had an option to keep such code advanced mode and make this filtering like flying in the air without landing anywhere in your system. But I personally think that this kind of advanced filter is not so good because once you take time to build it, once you make complex conditions there is a highly likely situation then you will need it to save. So developers in Freedom UI decided that no advanced filters anymore. If you need some complex filtering, just mentally prepare yourself to save it as a folder. It's not a big deal, not a trouble. And such folders usually have quite strict permission settings. So when you create a folder only your user will see it so it will not create too much

of garbage records seen by anyone. Only your user will see such records.



Speaker 1

If you don't like it you can also remove it. So I think it's not a problem. Thank you Adrian for your question. And yes I agree we have some changes between Classic and Freedom UI and it looks like such changes were discussed so it's not a real decrease of some important functionality. Thank you Adrian. So this example shows that Data Surveys is much more capable in comparison with ODATA because you can use more complicated calculations for columns for filters and everything is going with just a single query. How we did here and demonstrated in Postman that it works perfectly in real life. Your integration will include not but one or two queries in real life I think you will need tens or even more queries to start. So it means that you will have series of different queries.



Speaker 1

You will need to remember data and save it somewhere. And in general it requires some quite strong professional developer skills and architectural understanding of what you are doing. In general, I recommend you to run queries that will not return you millions of records. It's better to operate with data using some portions and both ODATA and data service have their own limitation. So we can find some limitation. Number of requests is unlimited but integration options. I saw some information about it that we have a limitation about 20,000 of records per one selection and the same limitation will be applied for data service. This is limitation for OData and data service has very similar limitation. But I just suggest and recommend you to operate with data with some smaller portions.



Speaker 1

Keep some logging by yourself so it will be easier for you to detect and understand how actually it goes and do you have any serious errors in your integration. So it's like a programmer task, but it's possible and in complex projects we also do this and it works well. So let's move on. We already studied how third party app can operate with creatio tools and odata and data service are already present at creatio server side so they are already prepared for you. So you only have to program at third party application to correctly call such tools and it will work for you. Now I wanted to show you an example how we can run third party web services from Creature and also how to do it with no code tools because no code is really attractive.



Speaker 1

It takes very little time to develop and gives quite quick and valuable results. Let me show you. So let's imagine we have Creature app, we have different sections, data and so on and we have some third party application that we want to call and to use its data to perform some data transfer. Let me show you some example of it. We have a node with REST API samples so we can use something like this. Let me show you. This is a REST API URL call that we can send. This is a GET query. You can easily do it in your browser. Luckily it does not require any authentication. So it's a GET query performing with some kind of URL. No initial arguments but we have some response.



Speaker 1

JSON body this JSON body is also shown here for us and this JSON body represents some prices and I need to tell you some physical sense of it. ID bank is one of the banks in Armenia and they sell gold bars, gold plates starting from 1 gram gold to up to 12 kilos gold, big gold slab and you see their prices are in local Armenian currency, Armenian drams. But in our case it's doesn't matter so we only want to practice here and also I want to show you some additional example. Not only just get couple of data values from third party app but also to show you how you can use no code tools to process collections and to get sets of some data records, how to store it, how to operate with it.



Speaker 1



So let's imagine we have a task to regularly get gold prices from this URL, from this API, save it in creatio and make it possible to run multiple times and correctly update such prices. I will use only no code tools for it and you will see how we can do this. Also you may reproduce it just during today's session. So let's start. We have URL and it's nice if we have some description of this URL. So generally we have it but in my case it will be really simple so I did not really need it. But normally when you work with some kind of integration you will have this kind of explanation of how to call corresponding query, how to transport some parameters rate go. Okay, now you see example of query, you can see parameters, you can see requests and so on.



Speaker 1

And in my case so I can also share with you this is kind of documentation but the most important we need example of call which works for us. Luckily we don't have any special protection, security or authentication necessary for this. You may also find a lot of other sources with similar functionality like currency exchange websites, like weather forecast websites, like other regular data that could be commonly interesting for people. It's usually shared without any special restrictions. In my case this doesn't require any parameters, any input arguments. So we can do it quite simple and I think that simple example is better for beginners just to get started. So we have this URL and then I will explain how we can work with it. First of all we need to go to studio workplace and find web services section.



Speaker 1

We will properly register our new third party web service here and then we'll be able to call this web service from our business processes with the help of call web service item. This will not require actual development skills, but it requires some engineering knowledge and understanding of HTTP queries and HTTP methods type and just a bit of understanding of JSON string and you will see that it's not very difficult. So let's do this add new web service when we Created system uses current package system settings. So when we provide an example of a full web service URL, it parses it and creates corresponding web service setting and tries to save it. We need to take care about the proper package to save. Now you see some phantom packages which we can't really find in our configuration, so let's ignore them.



Speaker 1

This is classic package because our current package system setting points to it. But my plan is to save our data to freedom UI really package. Let it be here. This URI is like a main part which was extracted from our URL and it will be used as a like base part of our web service and then we can have many different methods. Method addresses will be added to our main web service URI address. The code is generated by app so let's call it something like gold bar service name will be displayed and the code will be used for configuration saving into a package. Finally, we will have a special type of item saved directly to the proper package. So you don't need to care about how such setting will be saved in our configuration.



Speaker 1

This will be saved as special metadata directly to the package. And we have a method here, so let's save it first. Now you see configuration section and let's go to all items. You will see new type of item called web service and it's already saved in our package. So all you can do is to open its metadata. You will see some low level text definition of the setting like addresses and other settings and so on. But this is low level setting. You just need to remember that it will be saved in our package so you don't need to care about special transporting of it. Okay, we continue to do our setup. System was capable to automatically parse our URL and get web service main part and method part and you can see it's by default. This is a get method. Let's look at it.



Speaker 1

And in general one web service may have many methods so you can register them manually in order to use different functionality of the same web service. Get method the content type is JSON and response timeout is 5 seconds by

default, 5000 milliseconds means 5 seconds, no authentication necessary. And name is for display, code is for metadata to save and method address is the most important property here because it represents exact part of the URL which will be used to add and make full. So when we do this we will make foo method address in our we have a full method address in our lab service and this part was automatically parsed here. You can also type different types of Parameters like method parameter, method query value and also it's possible to transport additional data values depending on the type of request you use.



Speaker 1

If you use get query like in our case you only can use method address parameters inside of this URL. If you use post, it's possible to fill in post body request body and it helps to get much more possibilities to transport different data values as arguments to your web service. In my case I have no request parameters at all, but if you will do it something like yourself with different web services, you need to remember that we have all possible ways to transfer data that is commonly used in the REST API. You can use address parameters, query values, header values. Remember like we did in our post query special headers setting. So it's like technical settings that will be passed as part of our query. So creation is also capable to provide headers and even cookies could be provided separately.



Speaker 1

So if you do some operations with third party system which requires some authentication or requires some cookie for like your user settings, then it's also possible to use it in your query. So it's very universal tool. In my case I do not need request parameters in this example but I need response parameters. So let's save it again just not to lose anything. I will use response parameters first automatically. Let me show you. First of all I can run send test request, no parameters, no authentication so I just go and send it and I have a responsive JSON or in row HTTP. Of course JSON looks much more friendly for me and I will just copy this data and this is just an example of what web service answered me. Okay, this is my response then I go to the method switch to response parameter.



Speaker 1

So my idea is to tell creatio how should I parse result data in order to extract specific values from it. So in my case response parameters can be added manually or we can use special very effective tool which is called quick setup and I will use example of my response body and give it to system in order to detect what are the possibilities, what are the possible values that we can get out from this example. So I will use setting of response parameters with the help of example in JSON quick setup example of response in JSON of course I have to paste my example of JSON data obtained as a result from my test request. Then I click next System was capable to parse my data.



Speaker 1

As you can see it was it detected that I have two collections one is called cash sell it list of sell prices so bank sells gold bars and cash buy it looks like they have this data. But zeros tell us that possibly they are not really planning to buy anything. And it's just like useless part of the data for us. So we can select what part of data is interesting and what part can be skipped because in our case we do not need it. And this is a very good example because in my case I do not need cash buy part, but I need cash sell. Okay? And we can select only part that is necessary. It's important.



Speaker 1

This is very important because in real life examples, for example, you can call some foreign currency exchange rate service and you know, we have almost 200 of different world currencies and you may face maybe 4 or 5 or even 1000 of records, 4 or 500 or thousands of records with different values and parameters as response. So it will be really important for you to select only parameters that you really need to get from results of web service. Because sometimes web service result body is quite excessive, including a lot of information that you don't plan to use or just not useful for you. In my case, I am interested in cache cell list and I will save it. Okay? So this helped me to avoid manual registering of parameters. It's possible I can create such parameters manually. I can reproduce everything by

my hand.



Speaker 1

But using this quick setup tool by examples is much more efficient. So I have a root item here which represents a collection is array. And also it's interesting to mention that creature uses such thing called JSON path. JSON path is a kind of address of a value inside of a JSON body. So this value helps to detect and get corresponding value out from JSON body text and creatio is using it to allocate and find the corresponding parameters. As you can see, it's body parameter and inside of this array we have pairs of data weight and rate. Let's look at our data closely. Weight usually so it's shown as a text, but physically it looks like a integer number minimum is 1 gram and maximum is 12,000 grams, which means 12 kilos. So we can afford to treat weight not as text but as integer.



Speaker 1

Because we see that for all data values here it will be nice if we have an integer so we can treat results as integer and system will be okay with it. But when we go to rate and see some examples sometimes. So in general rate is a text and sometimes we see thousand separators and sometimes you see it even twice. So it turned out that this mechanism which gets data automatically with the help of data service works poorly with type conversion. That's why it makes sense to keep rate obtained as text. And if you really need to work with it, as with decimal, obviously this is a price, so it should be decimal.



Speaker 1

So if you really need so you can parse this data lately after you've got it from the web service and then later and then you can save it as you wish. In my case I keep it simple. I try to not spend too much time on such data type transfer. So I prefer to keep it as text. So it will be just a good for our demo. It will be not so suitable for real life calculations. But later it will be not a problem for you to use date type conversion using. For example, you can use script task for it. If really interested we can try to do this. But I'm trying to, let's say save our time and to not to go quite far from our data obtaining from web service.



Speaker 1

Okay, so this example takes result data and parses it and finally will present our data as collection of data records. We can save it. Save it. And this information is now saved in our configuration section. Now I propose to make a five minutes break. I promise not to go too far and then we will continue. So let's move on. And we finished on the creating of our web service. Checking out that our web service was saved into our package. This is very important. And now we will use it. So what how we can use it? We can create a business process which will run this call web service item. We will get response data. But we need to think of where we plan to save set of records with weight and rate information. We have only two possible options. One is memory.



Speaker 1

But it will require some C sharp scripting for us in order to keep data in redis memory. For example, I prefer to make it more simplified and no code. So we will save our data into our database in just a simple lookup. This lookup requires integer weight and text rate. We can easily create it ourselves in this package. So I will do it Add object. This is necessary to organize storage of our data. And this storage will be named USR gold price. No, we can do it parent object but not base lookup because we don't need name and description. I will use base entity because it will only give us standard parent columns and we can add business columns ourselves. So we will create integer usr weight colon. It will be integer and one more will be text 50.



Speaker 1

The minimum one usr rate will be our rate column text 50. You may ask me why text? The answer is because I do demo of getting data in real life. We will need to save our text values first or maybe to use them and process before saving with the script task and then we will use a decimal value to save finally converted value and in my case I went just to keep it as demo for you. So we'll save a text rate data as we get it from web service and that's all. No anything else we can publish our object. As you remember publish always performs save first so we don't need to click save button. Save was performed and publish was done. Great, now we have this object already applied.



Speaker 1

One more small step we can go to lookups and register this object as lookup. So we click lookups new lookup find our gold prices save it. So now we have this lookup we can open its contents but we have nothing there. We can open properties and create data binding item and save it into our package. This is necessary to remember registering of the object as lookup. This is mandatory step if you want to go to your lookups and find it when we will go when we will deliver our solution to test environment. So now my gold prices object is ready and we can move on to work with processes. So we registered our web service and we can use no code solution with processes to get this data from third party.



Speaker 1

We can create new process name it like USR yet road price main process this is main process because I also plan to use sub process to parse my collection. That's why I call this processes main process. No initial arguments simply starting our process by manual start and the first step we need to do is call web service process item and which web service to call. We have our ID banking web service then which method to call? We have the only one method that's why system selected eight method for us automatically. We don't have any request parameters. When we switch to advanced mode we can see response parameters and response is our collection with couple of values here also we will get HTTP status code which will be useful to check different errors and analyze what's actually happening.



Speaker 1

And sometimes full response body will be important because it will be it will include everything obtained from third party without any parsing Boolean success property and probably that's enough. And we also have a SO request body but it looks like this is an input parameter and only for very specific cases. So I don't know how to make a no code example with this response request body so we will not use it now. Okay, so when we run this call web service we can run it like get Gold prices. This item will finally run this query. Let's go to primary mode, get code prices and it's important results will be saved somewhere inside of this item inside of the response parameter collection. So we will obviously have to process this collection somehow.



Speaker 1

And first of all I will show you some error handling in case if our request finished successfully. We can turn this flow into conditional flow by clicking on this change type button and then set conditional flow so we can name it okay and we can check condition. This condition will include just boolean success property. If it's true then we will go here. Otherwise we will stop and we will have another terminate item code error and we can go from our call web service to this terminate item. You can name it error means something is not good and we can just turn our flow as into default flow. Default flow will be activated if no one from conditional flows worked for us and default flow will be our error handling.



Speaker 1

So if we not successfully code our web service, no need to try to perform next steps. We just need to stop our so abort our process and use a separate terminate item because it will be stored in our history of execution so we will easily understand that it finished with an error. Okay, but if everything works well, we need to think of the place where we plan to keep our data when we first run it. Obviously our data table will be empty so no visible preparation necessary. But when we run it next time our data will be not empty and maybe it makes sense to clean it with the

delete data item. So I will do this delete data and this will be an item to remove data from go price object. And as you can see we have kind of protection here.



Speaker 1

This protection means that for delete data we must use some parameter. If we don't, this will not allow us to save process and it will show us errors. So we must use some parameters here. And this is kind of protection from unintentional delete of all the data. But in our case we intentionally want to do this. So we have to perform some kind of fake query which will be always true. And that's how we can make this correct condition to delete data. So I will make a filter like ID is filled in for any existing record. This condition will be true and that's why I can use it. And it's quite easy and simple. It will not require too much resources from system to make it. So this will be an item like clear prices storage.



Speaker 1

So we delete prices from our storage and then we can use so we have a Collection and we have two options. To use no code approach with sub process to parse collection or to use C Sharp code in order to parse our collection with programming. Of course I prefer no code approach. Let's save temporary our current progress. Oh, it tells that the change is saved. Cyclic change in packet hierarchy. Probably it's because of I forgot to make dependencies and settings. Yes, it was my fault. Okay, it was unusual, but it was my fault. Well, it's because of current package system setting was pointing to classic UI package. I have to switch to realty. Yes, modify my setting and save again. Now I have no cyclic dependencies. Let's go and check our packages.



Speaker 1

So realty package should be dependent from dev classic and dev classic package should not be dependent from realty. Yes, looks good. Yes, looks good. Very good. So now no cyclic dependencies, no troubles. And this was just because of the package. So that's why. So we worked with process library. If you select a corresponding package, if you start to create your process from here, then your process will be okay with the correct package setting. So possibly we have one more plus for creating processes from configuration section, but not from the process library. Okay, we have first now let's do not repeat this mistake again and we will select our package. Add one more package. I need a sub process to parse my collection. The main idea how we can parse collection is to make a sub process and to use arguments as parameters.



Speaker 1

So we'll use parameters and I was a process to accept collection data values and I will make my process name USR add gold price subprocess. So it will be my sub process to add the gold prices. I need parameters because the only way to transfer data from main process to sub process is sub process parameters. One parameter will be integer. So I add the parameter which will mean weight integer input which make it makes it read only inside of my process. But I don't plan to change it. So I only plan to get it as input value no initial values. Save it. Another parameter is text 50 the shortest one and it will be rate and it's also input and no initial value. So here we have a couple of values we expect to get at the beginning of the process.



Speaker 1

Then we will use our process and our process structure will be really simple. We will just use one add data item. We will add data into gold price object add price and we will fill one record and only couple of values weight and rate will be filled in very easy rate and weight. So rate will take its value from corresponding Rate text parameter weight Clicking on this lightning button, we'll get its value from corresponding integer weight parameter. Please note select parameter window always filters available parameters according to the data type. So that's why we see only decimals and previously we saw only text parameters. Okay, we have this stuff. So you see the sub process is pretty simple and we can save it, close it, go back to main process.



Speaker 1

Now we will use an item called sub process and I will use this orange item and place it into my diagram. Normally when you want to run a single instance of a sub process, you just specify your sub process name and single instance call means you have to transport. They are just pair of values and this means that you will run your sub process only once. But in our case we plan to work with it and parse collection with the help of sub processes. So I will use special settings. Now please be careful and watch here what I will do now. For example, let's go with wait first. So we plan to turn this sub process into a collection processing mode. And this can be done by selecting special values here. And such values should be taken from collection.



Speaker 1

So I click on this lightning button, select process parameter and then I will select process elements and our get gold prices call web service. It returns collection and we have corresponding weight column in this collection. So I have to double click it. And that's how our sub process immediately turned into collection processing mode. And it will run as many sub process instances as many collection data records we have. And we have execution mode sequential or parallel. In my case, no need to run parallel sequential means one by one and we have input collection. We already specified weight column for it. Now we will set rate column from process parameter rate double click. That's how we do this. Run it in the background is not necessary here will probably only take some additional performance.



Speaker 1

But in here we can just run the process and we are interested that in finishing it fully synchronously. No strong reasons to run it in background creating some scheduler executed tasks. So we just can run it simply in a current thread. Okay, when it runs we can also call it like add prices. Okay, it prices and it will run add vote price sub processes and we'll have as many sub processes as many collection records you found. So that's it. Our example is ready. So it will be our happy finish item. This is our start. It makes sense to note and make names for all the items. It will be like a documentation which makes it easy to understand and we have current version. Okay, let's Save it. So now we have this process and we can start it from here or from process library.



Speaker 1

Let's reload it by clicking on this icon and we have to start main process. Let's start it. But also before starting it we can enable trace to be important for our debugging and we can also. That's it. Okay. Trace is the only one option suitable for us. So let's run this process. Then we go to process log. We will see our main process started and it took a bit longer than 1.2 seconds. And then you see number of subordinate sub process instances executed quite fast. But we have approximately 11 items here. So we can check main process and open its execution diagram so we can see how many times our sub process started. 11 starts, no errors and it looks like everything went fully okay. You can also see trace data. We are interested in how exactly our gold prices were obtained.



Speaker 1

So we can click on show trace data. You will see all the parameters in your collection. We can see technical parameters like response status, code 200 means everything is okay, full response body sometimes can be important. And it looks like in our example everything went smoothly without any errors. So this trace takes some additional performance. But in general it's a good idea. And this trace helps you to understand if something goes wrong. And you will see exact parameter values, exact status code. So I recommend you to keep trace on if you want to support possibility of quick discovery of something if something goes wrong in your integration. So in my case everything went well and we go to lookups to see exact prices. So code, prices section, lookup list and we can just display weight and then we can display rate, save it.

Speaker 1





Also it makes sense to look at the columns of creation, date of creation. Okay, and now you see we have all obtained data saved in creation. Physically this is text. So if really needed we can use parsing. We can use some C sharp scripting to parse such values into decimal and then save them into corresponding decimal fields. But technical side of running queries and asking third party system and to return some data. I think it's quite clear in case if you have any questions, feel free to ask, I will be happy to answer. This is example how we can call third party apps. And at the beginning of course you will need some example of a call some URL, example of parameters, maybe some description and documentation how to call this password public API. And you have all steps made with no code.



Speaker 1

And finally we have a process which takes probably less than hour to develop with all the explanations and we successfully got our data. If we start this process one more time, you will see in our lookups, you will see new data. So you see date in time was changed. So our contents of previously obtained records was successfully removed and we have new data records inserted just seconds ago. This is also important when you develop your integration to check how it runs multiple times, because if it runs once, it's okay, but you should expect it to run regularly and it should properly include the existing data and correctly operate with it. Probably removing it, probably updating it. It's up to you in case, if you have any questions about it, tell me please.



Speaker 1

My next topic for today is to explain you how to use Clio tool and explain why you need it. So first of all I should explain why we have some additional tools and not inside of the base platform. So previously we already had a tool that was necessary to support developers and this tool was called Let me show you Delivery Tools and Workspace Console Overview. So previously we had a special tool called Workspace Console and it's probably more than 12 years old and it was used only for developers and only for developer tasks. Sometimes tasks were included based product preparation, processing resources, translations and so on. And finally it evolved into a complex tool with Many more than 50 different functions related to C Sharp sources, related to stored localization files, XMLs working with file system, working with version control.



Speaker 1

And so it has a lot of different, let's say quite technical purpose, which probably will be not necessary for end users. So developers used it for a long time. But now this is an old tool which has its own disadvantages and developers wanted to make something new, so they made a new tool called Clio. You can find it in GitHub, ATF Clio. If you search for this, you will easily find GitHub repository. You can find its root folder and then you can see a list of owners or maybe authors or contributors into this tool. And you need to remember that most of them are Creature employees, only some of them are outside of a company. So you see some strange users here and there are some people from outside of Creation, but most of them are from Creature.



Speaker 1

So generally you can consider as this tool was written by Creation in general, but this tool has open source code. You can easily not easily, but you can analyze its source code. You can like propose your own improvements into it. You can even add your own functions and also collaborate and helps to fix maybe some bugs and maybe to improve documentation and so on. So this tool is open sourced, so it's free of charge and you can use it at your own. This tool has also more than 50 different columns comments and I need to quickly show you how you can use it and why you may need it. I will focus on the most useful examples that you may face when you develop your projects.



Speaker 1

But you should keep in mind that this tool was written by developers of base product, so they had their own reasons

and needs to operate inside of base product. And now you will see this difference. So let's go and I will show you how to use it. This tool was designed to use by command line, so you should use some Windows or if you use other this other operating system you should use command line for it. And first of all we need to check presence of this tool and you can find documentation here how to install it. So let's let me show you. You must install dotnet core framework on your PC if you want to use Clio.



Speaker 1

I already have it dotnet tool list G this is command showing me existing installed dotnet core tools on my PC and Clio is already installed here. Okay, I can remove it dotnet to uninstall Clio G It means remove it from my system. Now if we check we have no creature Clio tool on my PC. This is a common line utility. It's managed by command line parameters and the most important property of it. It operates with target creation system with the help of web service calls. So Clio is managing creation by help of web services. It means that it can manage local or remote instance the same well so unlikely to workspace console which required file system access and database access to your creation. Clio requires only Internet network access to your creation.



Speaker 1

Okay, let's install it.net sorry to install Clio G this is command that you can type in your windows command terminal. This is two this is common to install Clio. It will install the latest version. Let's see what version we will have. Now of course this comment requires Internet connection so it goes to nuget searches for the last package which it can find. And now you see the Last version is 61016. So this is current latest version of Clio. Okay, we installed Clio but not set it up fully. We already can run Clio command to see plenty. Oh sorry, my fault. Plenty of functions that it can do for you. So let me show you where is my scroll bar Here you see more than 60 different functions and I have.



Speaker 1

I have suspect that the most early created functions are at the top of this list and the most recently lately created functions are at the end of this list, you will not need all of them. So it's like a universal tool. It has a lot of different parameters, a lot of different comments. So I will show you the most important, the most practical that you may need. And after installation we need to tell Clio what environments we will work with. Because Clio operates with target website and it needs to know URL, login and password. So we can use command clio show webep list and it will show a list of registered applications that are already present on my disk and physically saved into this app settings JSON file located somewhere in my user folder on disk. You see my previous attempts here.



Speaker 1

So I already have some previous environments registered. I have to register my current environment. Clio reg web app I need to show you reg web app. Of course you need to know how to spell certain comments, how to get their parameters. So you can try to do something like this. And for some comments. Oh, doesn't show us. Okay, let's do this like this. So for most of comments you can type minus question and you will see some error that it did not obtained, did not understand the command. And it will show you a list of all supported comments. Most of commands have their short version and most of parameters have their short version. So in order to save your space you can use shorter parameters. And the reg the app command is used to register your app.



Speaker 1

So it will be included into this file and then you will refer to it by name. This is easier than providing URL, login and password each time when you need to operate with target environment. So let me register my current environment. Clear reg the web app. I already have some example, so let me type it. So I have D1 to do. Yes, this is my environment. Okay. L means login supervisor, P means password supervisor and my name will be D1. So this is how I do my register. Copy it for you so you will remember it and you will understand why I need it. So when I register my app. Oh, we have some additional troubles. And path environment list is net core. So we have some troubles in existing app settings. JSON okay, it was not expected. Okay, show the list.



Speaker 1

Wow, we have some troubles in this file line 41. Probably because I have some older versions of such settings. Okay, let's go where it tells users. What else? Users or user app data Local creation My user app data local creature. Sorry, sorry. Local creation Clear App settings JSON possibly I have some incorrect settings here or maybe outdated settings. Oh, I have something like this, which obviously it doesn't look good and probably it's a result of some previous stuff. It looks like I have some incorrect settings here. So okay, environments closed. Then here it's end of environments. Here is end of all stuff. Also I could remove my previous items here. So I have no environments. Now let's try to ask it. Clear show web app list. No troubles. You see, everything is okay. Clear reg web app. Configure it correctly. Okay, great.



Speaker 1

So now we can do some simple operations with it. Cleoping and then we have different options. But the easiest way is to use E key for environment and then environment name. So this cleo ping will physically check availability of our web service. And if it's okay, you will see yellow. Sorry, not yellow green color text. So ping was successful. One of the functions that can be useful is restart. So you can use restart function. Restart comment clear restart ed1 this is for restart of our app. You remember we used maintenance tools add on for this. But also you can use this command to perform restart. Usually it's safe and quite quick. You will see what is happening. Now our application is being restarted. It takes up to 10 seconds and sometimes you will need this restart as part of your development process.



Speaker 1

You remember when we modified some objects, added new columns, we had to restart our app to properly apply such changes. So we had such scenarios in our training when we needed it. And simple log out and login did not help. Okay, so let's move on. We already started restart. We have a lot of other commands and you will see plenty of them. A lot of comments. I will show you the most important ones. So from project development point of view, you need to know that Clío can really help you with saving packages and loading packages. So you can use Clío to download packages. This one, this comment. Download packages from source environment. For example it can be developer environment and then you can use install package. Let me find it. So we have download and we should have upload. Push pkg.



Speaker 1

Let me switch find it should be somewhere here. Yeah. Push pkg. It has also short command install. I don't know why developers did not show it here. Install command is capable so it should be shown here. So you see documentation probably is not perfect. And the install command loads our package to target environment. This can help us to organize CI cd continuous integration and continuous delivery with a single script that can be started with the one step one operation. We can download and then we can install it. But also you need to know that create Clío has more other columns. Sorry comments and some comments like we use for download or install are using standard base product creation Web services. So system that we work with doesn't require any special setting.



Speaker 1

But if you want to use all power of existing Clío commands, you must perform special command and this command is named Install gate. This Command downloads from NuGet and installs in your target system special package which keeps DLL inside. And this DLL includes all necessary web services that enable this set of commands. So such web services, simply speaking, are parts of Clío to implement corresponding comments without this comment, without install gate, without additional package installed to your system, CLIO will be not capable to perform corresponding commands. It will show you error 404, which means missing functionality in your target environment. So Clío only runs web services at your creature. And if it fails to run web service that it needs to perform a corresponding command, you will see this error. So let's try.



Speaker 1

Let's try to run a command to install gate Clio install gate ED1 this is required in case if you want to enable full power of Clio commands. So in my case, I will do it in my source environment. This takes time because it Downloads package from NuGet, it installs this package into your target environment. This package is called Clio Gate. So you will see new package Clio gate in your system. And this package will have so called file contents and file contents will include compiled classes for your web services necessary for Clio to work. Let me show you. So here we have configuration section. Okay, it looks like we have some stuff here. Let's reload the package. Reload configuration section. We have some interesting information here. I did not expect it. It looks like we have some troubles with all data for compilation.



Speaker 1

So system tried to compile and failed exceeded narrow try count work process. So our IIS prevented our system to compile. Let me check. Did I detach? Of course, my visual studio was detached. Okay, I need to look at such questions. And also we see some troubles with old data and probably it's file system issue. Okay, so now system tried to install and compile package named Clio Gate. I should expect it to see it here. No, now sort was performed with this case insensitive sort. Clio gate is a package with functions that are necessary to run Clio commands. And you see only a couple of data items here. Of course, it's not full stuff. So we go to file system, we go to our application, go to our packages Clio Gate package.



Speaker 1

You can see its size 5 megabyte of executables not so little files Bin and here, here and here are DLLs included as file content for this Clio Gate package. So such DLLs are used to execute Clio commands. They implement the implement web services, install such classes into creature server side. So that's actually how Clio may work on a certain environment and how Clio may create and implement new comments. Because if you want to run something, your target system must be able to do this. Okay. So if necessary we can fight with it. It looks like I have not so much time today to fix this configuration issue. My command to install Clio gate package was quite correct. So it looks like everything is okay. So now I can show you some examples how you can use Clio efficiently for project development.



Speaker 1

And this will be comments to save and load environment settings and Sorry, Save and load packages. Let me show you. I already have another test environment. You probably forgot about it. So we have D1 Studio as development environment and D2 second was used as a target and the test environment. Let's check it out. It should be alive. So let's go D2 it's used as test environment. So why it's loaded? I can also use Clio to. Sorry to register a new environment. So we load it here and then we will use it. So clear rag web app. I need to register D2 and I will call it D2 test to make it clear that it will be my test environment. Okay, register it Clio show the back list. Okay, my D2 test and Clio ping P D2 test. So.



Speaker 1

So I will check availability of my second environment and it looks like should be okay. Yes, looks like Bing is okay. Here is my second environment. So my plan is to show you the most useful usage of Clio for project development. This is CI CD automation for saving and loading of packages. I already have examples of scripts that perform this save and load. So let's take this one. Probably Clio download so we can download set of packages. Let's check out how many source packages we have. So we have Dev Classic. Okay, here Dev Classic. Then we have USR Realty and also we have USR Realty migration. So we have three packages here. USR realty migration and Source environment is dynamic one. So Clio download is the same as Clio pool PKG command and we can download set of packages.

Speaker 1



Not only one but several packages from source environment. Destination path is somewhere like this guided dev. Okay, let it be May 2024. So we will finally have a result of a zip file including all our packages. We can use this zip file into next command which will be used for loading clio install the same as clio pool push pkg we will use source file I will use on target environment D2 test and we need to use logs because it's important to see some technical details. So I will share such comments for save and load with you. It will be really important for you if you want to practice with it. And now let's check out how it works. We will remove unnecessary examples and this will be my save and load example of CI cd.



Speaker 1

Let's try first system will save three packages into gz files organized into one single zip file and then it will be done and as you can see it makes it as a one operation if you do save manually saving of an app or saving of a separate package. So you will have to perform it with separate structure steps. As you can see it went but not so fully correctly. I will remove unnecessary files from my example the 7th of April so those are here. So here is our saved zip file. So it looks like save part went okay and may part it looks like I had couple of exceptions here. Let's analyze our logs.



Speaker 1

So you see we have location packages more file descriptor more than one file app descriptor JSON what it is what package it's about real to migration No, I don't know. So something wrong with application descriptors we can check it at our file system. Probably it's a result of something messing with package dependencies. So our realty has its own information of app descriptor yes, this is realty app. Okay, realty migration has its own app descriptor yes and our dev classic should have no dependencies from existing realty stuff and it should have oh we have app descriptor extension.



Speaker 1

Something is strange here and it looks like probably system failed to work with it but it looks like it's not a big trouble and you see one more issue here parent schema and was not found yes, this is more serious trouble because it looks like our target environment was not fully prepared to install our changes because it miss customer 360 app possibly previous error message that we had here previous message with this stuff here probably it was also missing corresponding app and we used so our solution expects that target system will help customer360 in my example customers current our target environment did not have customer 360 that's why we have this stuff. So in order to fix it of course we have to install customer360 first and then our solution next. In this case it will be loaded fully correctly.



Speaker 1

But now we can test our target environment. So usually we need to reload it. We can find our items like oh, we don't have any, probably did not like it. Okay, we have to install customer360 here and then load our solution. So you see only Dev Classic was loaded here and recent stuff was not loaded. Okay, let's try to install customer360 install. Continue. I did not enable file system development mode here because it's like a test environment and no any creation, no any external ID. So let's install it. I install customer360, then I can try to load my settings again. That's how we will see our working example. So I plan to show you fully correct transition of my environments. This is customer360 installation on my target system. As you can see previous results of manual load of package.



Speaker 1

Manual load can be done with new application and then install from file. Now we have successful installation. Great. If we reload we will see customer360 app added. Yes, customer360 was added. Now we can try to install our stuff

one more time here. Let's remove previous package. So we will. And one more thing before we continue. So this is not so obvious, but when we developed C sources we used external editors. We have C code saved on disk only. And in order to make everything correct, we need to download all items that we develop manually inside of our embedded editors back to disk. And then we have to upload changes from disk. This is very important because our database part of C Sharp sources may have old C source.



Speaker 1

And if we ever try to compile our package on target environment, we may have very unpleasant situation when we will try to compile old contents of C sources instead of actual one. So in order to make correct export of your solution to test in production, we need to save everything on disk. Then we have to take everything from disk update packages from file system. Now you will see system will show us differences in C Sharp code. Because both examples were written here and here were written with the help of external editors. This is very important. Now our system is ready to export. So let's do this again. Save and load. Thank you Adrian for your time. Yes, we are almost over. We will continue tomorrow with preparation for developer action and also answering your questions.



Speaker 1

It will be not so hard and thank you for your time. So now I just plan to make it correctly with save and load and then if everything goes well, we will see our Apps we will see our functionality loaded to our target environment if something goes not as planned. This is also good because we will see some exceptions. You will see how we can troubleshoot them. It's important to understand how you may do some fixes when your package is not installed as you plan. Now you can see. Oh, do we have any errors? Yes, we had some errors here and it's interesting because it should not be so we should not have any issues here. Possibly it's about data. Let me check here we have a log file and we have data insert issue.



Speaker 1

So sys module entity USR really no then this module in workplace studio so violation of primary key. Oh, it's because of data binding which does not keep correct IDs but duplicate cases module in workplace. So it's registering of a section in the workplace and this duplicate is possible because of the way how data binding saves it. So we have one data item failed to load but in general okay, one more data item failed to load so we have only troubles in couple of data items. But general data was loaded so we have as a result some errors. But this is because of something is not exactly the same as our original stuff. Okay, let's try to reload our section. Let's see do we have anything loaded Studio my applications we have realty section.



Speaker 1

We have classic we miss reality sections here because such data items were not loaded. I will show you how you can fix it. Let's go to advanced settings. Let's find our new packages loaded here so we can search for realty package and we can see couple of data items. So let's see only status has error or needs to actualize here so you can see list of items that were failed to install Couple of SQL scripts from base product we don't care and only one item in our reality package was not loaded. We can see error message text and properties. So violation of primary key. It looks like system can't find the correct value of what and it's hard to see. Okay, let's see. So it can't find correct duplicate value of sys module in Workplace.



Speaker 1

It looks like we already have sys module in workplace ID and it looks like this data entity was not good. So we have some items here and we have key which was not ID and key was used by section and workplace. So we can fix this it's not good and this is behavior made by data binding tool. I don't really like it. So we can fix it with the another data Item organizing key by ID that will prevent the data from inserting twice with the same id so we can fix it. Let me show you. Since module in Workplace in Studio so it's our reality section. This is a data item. This module in workplace for Studio and such settings for key is not good ID key is better position for update if in case if record was



found.



Speaker 1

Great, and we can save it. And we have a lot of warnings, but it's about existing date. So let's actualize it save it. Just a lot of warnings about reorganizing of the same existing set sections that are already present in data. And also one more thing which we potentially have the same trouble sys module and workplace for this realty and it's 26 my application. Also it makes sense to our ID is already set here. Okay, great. Probably you can put position updates choice data save it. Great, Everything is okay. So we fixed it. We have to save everything on file system just to make sure we will correctly have all the data well in our repository version control. So now you will see that it's possible to save and load packages again. We can prepare something like order like v1 plus old file.



Speaker 1

Now we will have a bit better version of it. Okay, saved. Now let's try to run this transport again so it will be final one. And this is example how you can automate delivery of your changes. How you can take data from developer environment and automatically save and devote it to some other task target test environment. Of course, as developer you understand that sometimes things are not going very smoothly. Sometimes we have unexpected errors. And having such errors during the training is also good because it helps us to see how you can troubleshoot, how you should search for error details and how you should do all changes. So now you see next time we load it took significantly less time because loader is smart.



Speaker 1

Loader text takes zip file, takes information about its items, all the packages, all the items of configuration and it analyzes date of modification. If date of modification of the loaded item is the same as date of modification of items that you have already in your target environment, then it just skips it. And that's why if you have a big solution, but you have tiny small changes, they are from recent upload that you did previously. It will only analyze changed items when loaded on target environment and final total installation time will be small. And now you see only couple of data items that were failed were now updated and fully applied and everything else was correct. So now our system in our target environment has no errors. As a result of installation we can go to all packages to see items.



Speaker 1

Actually only base product stuff. Okay, don't care, no errors. Okay, so our package with realty was fully correctly voided. All the items that we have were loaded well. And we can test it at our user interface. So you can go to corresponding sections, you can try to create data records. You see all three actions, all three sections here. So our realty with columns you can create new data. So find me cat here. And you see now we have default values working. We have some test data. Test price, negative price, validation works too big price will also work and system will not allow us to save because of server side. Okay, like this. But now you see price is more than 1 billion. So everything is working as expected.



Speaker 1

So that's how you can deliver your solution to test and check it out and perform corresponding full scale test for your system to check how it works. Now you see details were filled in. So everything works quite well. And that's how you can do some automation with Clio. Clio has also a lot of interest in other comments I will show you just one of them. Clio SQL select name for contact. Okay, something like this. So this is example for Clio SQL execution. Sometimes when you work in cloud with cloud environments you just can't connect to the database directly and operate with it. So sometimes you need to run some simple select queries in database to make sure you have specific data. And that's how you can do it with your Clio.

Speaker 1



So this is example of how you can read list of contacts from your database. Of course your queries could be much more complicated. And when you run SQL queries please be extremely careful because SQL statements is the way how you can easily damage your system, how it can easily destroy your database. So you should be extremely careful and please check your queries before execution. SQL can really help you and to see some data directly. Unfortunately Creature has no tools to work with SQL easily from user interface. Now previously we had such tool but was designed by some Russian partner. And as you remember, as you know, Croatia gave up any work with Russia or Belarusian partners and customers. So we also removed all of the add ons from the marketplace.



Speaker 1

And that's why now we don't have any fancy good looking tool to run SQL query from application user interface. But now this is something that you can do. And the last information that you need to know about Clio is that it's also possible to use add on for Visual Studio named Clio Explorer. So Clio Explorer Is an add on that you can use in Visual Studio code. It automatically loads your information from file with connections for existing solutions. And you can use a lot of Clio commands from user interface including SQL. So you can do something like this. Then you can run your query. Let me see where there is a button to run query. This one. So this is Clio SQL comment and it tries to run my query. But what? Oh, D1. I already connected. Probably I connected. Yes. What's wrong there?



Speaker 1

Split editor. More actions. Oh, probably this. Yes, it was case sensitive. I don't know why it's case sensitive. But now you see some data we got from the database and you have some kind of UI stuff. But with the help of Clio Explorer. Clio Explorer uses installed Clio Explore tool. So without Clio installed it will not work. But this is user interface feature which can help you to work with a particular environments. Would you like to current. No, no, no, no. Thanks. So we have a lot of functions of Clio the most commonly used and some of them. And also you can use Clio commands from terminal window of Visual Studio the same as we used in other Windows console bars. So, thank you for your time today. Today our session is over. We will prepare for the exam tomorrow.



Speaker 1

As usual you will receive video recording of today's session. If you have any questions, ask today or prepare questions for tomorrow. Because we will have more time tomorrow. Thank you very much for those who stayed for this moment. You will see videos soon. Thank you and goodbye. Hi Dimitri, it's boss here. Yes boss. Please. Tomorrow I'm. I have other obligations to attend to, so I will probably follow it by video. But is there any special action I have to take the fast track on development certification? Oh yeah, I will explain. So I will send you final email with all video recording with homework for your fast track certification and quick explanations of what you should do.



Speaker 1

So if you agree to run Fast track certification, you should just respond to my email and we will arrange individual date and time for you to run your exam. Your exam will include check of the homework. So you should prepare homework before exam starts. So you will have a couple of weeks to do this and you should prepare for the test using self assessment tests at Academy. I will show and so at the exam we will look at your homework and you will run your online exam test. Okay. If you fail, you can run this test again later. So don't worry. It's a bit nervous. I understand. And Fast track certification is free for our guided learning participants. Okay, cool. So you will receive all necessary stuff and videos explaining how to prepare for the test, how you should answer questions, and so on.



Speaker 1

You will see the type of the homework Raki is asking will include. So let me quickly show. I planned to show it

tomorrow, but okay, if you ask now. So let's do this, I will quickly show you. So the type of the homework will include making your section, this one, making your detail, make programmed validation, some calculations, web service, and for those who want to run an advanced level, some additional business process and adding data records and some automatic update. You can use live Update to automatically refresh your screen. Or I will show you how you can use WebSocket messaging if you prefer. If you prefer to do it a bit more professionally. So the homework simply is just similar to what we did during our sessions. And it requires some programming for validation, calculations and web service.



Speaker 1

So it requires some javascripting and C Sharp scripting. And so it will not take too much time for you, I think, and I hope that it will be clear. So we discussed all the steps how to do this. If you will have more questions, you may ask tomorrow. So Rakhi, I think that those who passed our sessions watched our videos and practiced with their own. You can use even your own example because the name of the section is the same set of columns, very similar. So you can use your training session environment to perform your homework on it. So it may be helpful. You do not need to create a separate environment for your homework. Okay, thank you for your time today.



Speaker 1

Sorry for staying a bit later than usual and if you will have more questions, so prepare for tomorrow, we will have time for this. Thank you very much for today's session and goodbye. You will receive homework assignment tomorrow as well as all video recordings and all the questions I will answer tomorrow. Thank you and goodbye.