SRPCE

NAME:- Akash R

ROLL.NO:- 2

REG.NO:- 422021104002

SEMESTER:-5

DEPARTMENT:-CSE

SUBJECT:- cloud Application development

SESSION:- 2021-2025

IBM Cloud Object Storage is a highly scalable, secure, and cost- effective object storage service that can be used to store any type of data, including images, videos, audio files, documents, and more. It is ideal for storing large amounts of data that need to be accessed frequently, such as data used for analytics, machine learning, and web applications.

- To use IBM Cloud Object Storage, you first need to create a service instance. You can do this through the IBM Cloud Console or the IBM Cloud CLI. Once you have created a service instance, you can start uploading and downloading objects.To upload an object, you can use the IBM Cloud Object Storage SDK for your programming language. The SDK provides a simple interface for uploading and downloading objects.

  Here is a sample program in Python that uploads an object to IBM Cloud object storage

IBM Cloud Object Storage is a highly scalable, secure, and cost- effective object storage service that can be used to store any type of data, including images, videos, audio files, documents, and more. It is ideal for storing large amounts of data that need to be accessed frequently, such as data used for analytics, machine learning, and web applications.

- To use IBM Cloud Object Storage, you first need to create a service instance. You can do this through the IBM Cloud Console or the IBM Cloud CLI. Once you have created a service instance, you can start uploading and downloading objects.To upload an object, you can use the IBM Cloud Object Storage SDK for your programming language. The SDK provides a simple interface for uploading and downloading objects.

  Here is a sample program in Python that uploads an object to IBM Cloud object storage

IBM Cloud Object Storage is a highly scalable, secure, and cost- effective object storage service that can be used to store any type of data, including images, videos, audio files, documents, and more. It is ideal for storing large amounts of data that need to be accessed frequently, such as data used for analytics, machine learning, and web applications.

- To use IBM Cloud Object Storage, you first need to create a service instance. You can do this through the IBM Cloud Console or the IBM Cloud CLI. Once you have created a service instance, you can start uploading and downloading objects.To upload an object, you can use the IBM Cloud Object Storage SDK for your programming language. The SDK provides a simple interface for uploading and downloading objects.

  Here is a sample program in Python that uploads an object to IBM Cloud object storage

```
import io
import ibm_boto3
# Create an IBM Cloud Object Storage client
client = ibm_boto3.client('s3')
# Set the bucket name
bucket_name = 'my-bucket'
# Set the object name
object_name = 'my-object.txt'
# Open the object file
with io.open(object_name, 'rb') as f:
    # Upload the object
    client.upload_fileobj(f, bucket_name,
object_name)
# Print a success message
print('Object uploaded successfully!')
```

To download an object, you can use the same SDK. Here is a sample program in Python that downloads an object from IBM Cloud Object Storage:

```
import io
import ibm_boto3
# Create an IBM Cloud Object Storage client
client = ibm_boto3.client('s3')
# Set the bucket name
bucket_name = 'my-bucket'
# Set the object name
object_name = 'my-object.txt'
# Download the object
with io.open(object_name, 'wb') as f:
    client.download_fileobj(bucket_name,
object_name, f)
# Print a success message
print('Object downloaded successfully!')
```

IBM Cloud Object Storage also provides a number of features that make it ideal for enterprise use, such as:

High availability and durability: IBM Cloud Object Storage is designed to be highly available and durable. Objects are stored in multiple data centers across the globe, and they are replicated multiple times to ensure that they are always available.

Security: IBM Cloud Object Storage uses a variety of security features to protect your data, including encryption, access control,and auditing.

Compliance: IBM Cloud Object Storage is compliant with a number of industry standards, such as HIPAA, PCI DSS, and GDPR.

Overall, IBM Cloud Object Storage is a powerful and flexible object storage service that can be used to store any type of data. It is ideal for businesses of all sizes, and it offers a number of features that make it ideal for enterprise use.

Custom/user-defined responses

In addition to rendering HTML content in responses, web chat can render content from your own HTML, CSS, or JavaScript on your page, using a user_defined response type. This allows for a much better authoring experience for development, and it enables you to make changes to responses without editing your dialog skill. These user_defined responses also inherit styling and helper CSS classes from web chat, just like text responses with HTML. You can even make use of portals in advanced frameworks like React to render content from your main application.

To learn more about the user_defined response type, see the tutorial or the React based carousel tutorial.

Any CSS you define, either for HTML in a text response or for a user_defined response, should be prefixed by #AWACContainer.WACContainer. Web chat uses an aggressive CSS reset in order to allow embedding on your page instead of in an iframe. In order to override our resets, you need the CSS specificity provided by using the prefix.

Example

```
#WACContainer.WACContainer .myAweso
background: hotpink; color: indigo;
}
```

A common way to wrap CSS in #WACContainer. WACContainer is by using SASS or similar CSS templating engines. It is important to note that when SASS compiles your code, globals like @font-face, and @keyframes are not correctly processed when wrapped in a parent selector. In these cases you must manually pull these declarations out of the #WACContainer.WACContainer selector.

Do not do this in SASS

```
#WACContainer.WACContainer { /* Will
NOT compile correctly. */ @font-face
{ font-family: myFirstFont; src:
url(sansation_light.woff);

/* Will NOT compile correctly. */
@keyframes example { } from
{background-color: red;} to
{background-color: yellow; }

/* Will compile correctly.
*/ .myAwesomeClass { background:
hotpink; color: indigo;

}

}
```

Do this in SASS

```
@font-face {
  font-family: myFirstFont;
  src: url(myFirstFont.woff);
}

@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}

#WACContainer.WACContainer {
  .myAwesomeClass {
    background: hotpink;
    color: indigo;
  }
}
```

Languages

Most of the content displayed inside of web chat is content that comes from IBM watsonx Assistant and is displayed using whatever language that content is written in. However, some content displayed in web chat is static text that is hard-coded inside of web chat. This includes things like the "Type something..." message that appears as the placeholder text in the input field, the "Choose a date" text that appears on a date picker or the word "Suggestions" that appears in the suggestions menu when web chat provides suggested messages to the user. By default, these texts are displayed in English but the language of those texts can be changed. Web chat provides a number of out-of-the-box translations for these texts and web chat

can be configured to use those languages. Note, that this language is independent of the language you may have set in your assistant. In addition, the individual texts may be changed if you want to use different text or if you want to provide your own translations for a language web chat does not have support for.

The list of languages and their language codes that web chat provides out-of-the-box translations for are:

- Arabic (ar)
- Chinese (zh)
- Chinese/Taiwan (zh-tw)
- Czech (cs)
- Dutch (nl)
- English (en)
- French (fr)
- German (de)
- Italian (it)
- Japanese (ja)
- Korean (ko)
- Portuguese/Brazil (pt-br)
- Spanish (es)

You can switch web chat to use one of these languages by calling the updateLocale instance method and passing it the given language code.

Examples

```
<div class="ibm-web-chat--card-home
<div class="ibm-web-chat--card-home

<div class="ibm-web-chat--card-home
  <div class="ibm-web-chat--card-ho
</div>
```

Using namespaces

If you are using the [namespace](#) configuration option, you will need to run the following command in your browser console to get the correct unique hashed CSS safe `id` for your namespaced web chat.

```
document.querySelector('[data-names
```

This should output an `id` that looks something like `WACContainer--ns-308302247`. You would then use `#WACContainer--ns-308302247.WACContainer` as your CSS selector instead of `#WACContainer.WACContainer` for all styling you do with web chat.