v5 Using Templates for Standard Processes

The Story

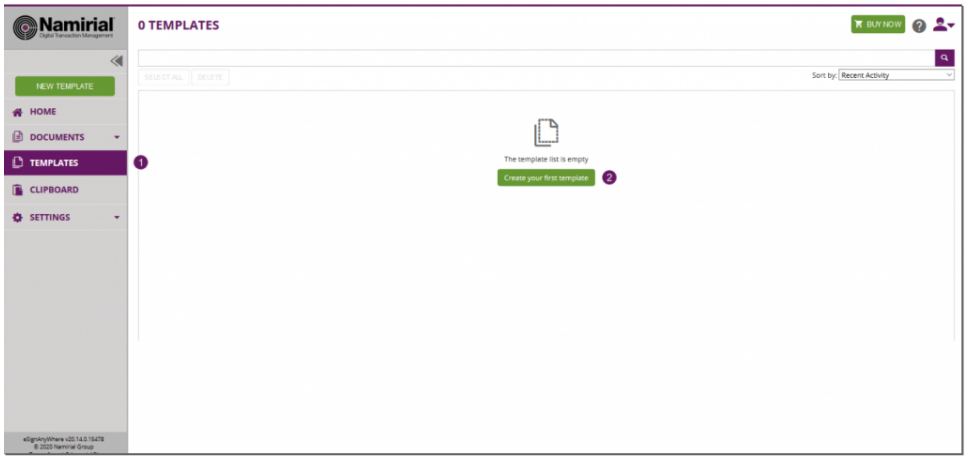
In order not to have to create the same envelope again and again for constant situations, the seller creates a template that he/she can use for constant contract. To create a template and use it in constant situations does not only save time but also prevents mistakes such as for example a text field being forgotten.

Use Case Description

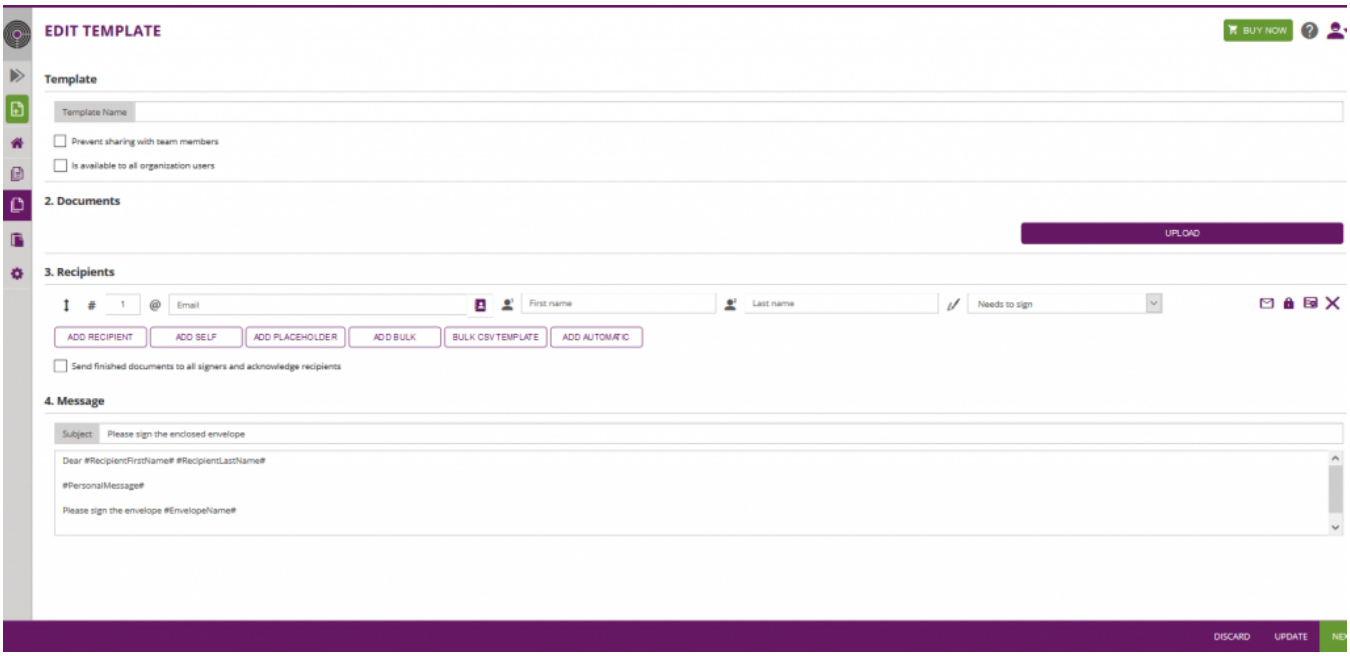
Following steps are necessary:

1. Create a template
2. Use the template

This case leads you through the process of creating a template with the UI and working with the template with the API. In order not to create the same envelope again and again this case explain how to create a template. Thus, time can be saved. First, we have to create a template with the UI. The following screenshot shows where you can find the template section:

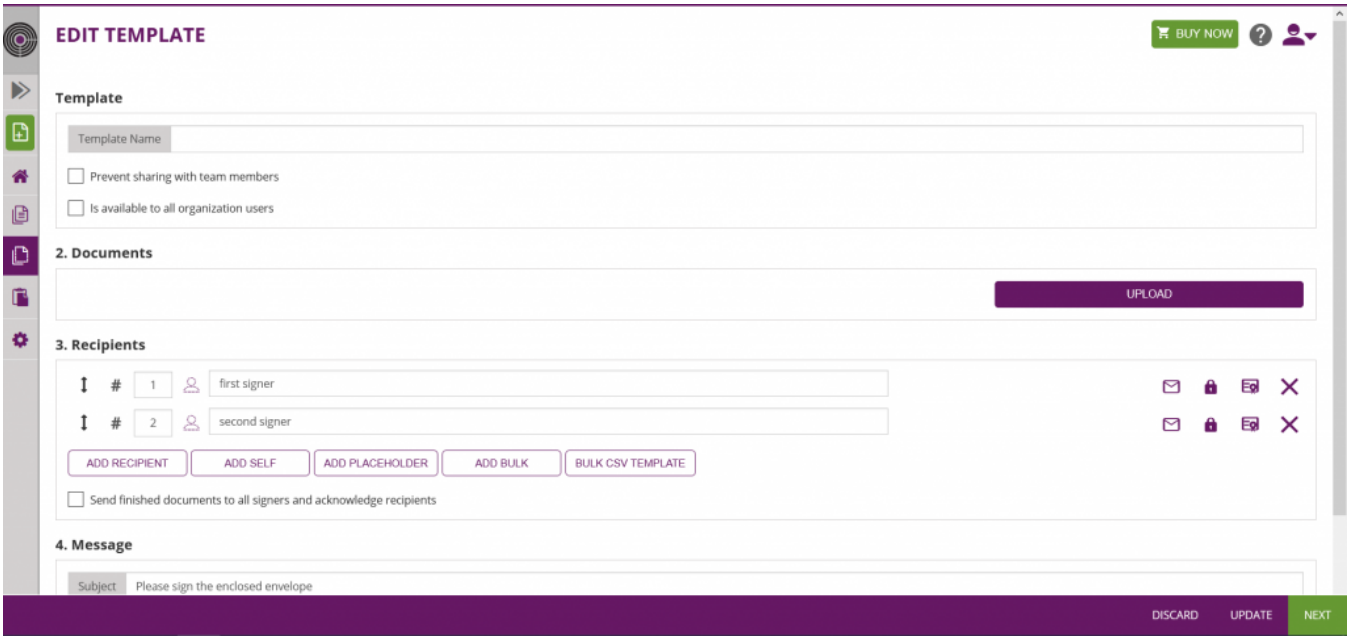
Figure	Description
	1. You can create a new template in this section 2. Click this button to create the first template

If you click on “create your first template” the following window appears:



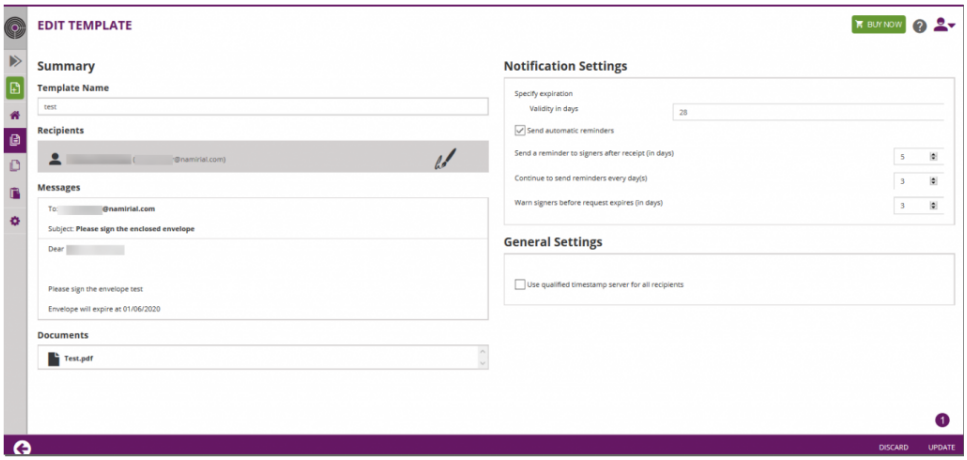
You can either define a recipient for the template like it is shown in the figure above, or you can define placeholder. Note: If you define placeholders for your template you have to override them if you use the template.

Here you can find a figure of a template with two placeholders:



There you can configure all settings for the template. After the configurations you can update the template which is shown in the next screenshot:

Figure	Description



1. Update and save the template

After saving the template you can now use the Id of the template and work with the API. If you open the template you can see the template in the URL:

e. g.: https://demo.esignanywhere.net/Templates/Detail/##TemplateID##

Integration

Note: If you have defined placeholders for the template you want to use with the API, you have to override the email address of the placeholders.

To get the email addresses of the placeholders you need the get envelope API call (envelope/##templateId##). In the response in the section steps you can find the email address of the placeholder. Afterwards you have to override the email address in the sendFromTemplate API call.

Please see the following configuration:

```
{
  "TemplateId": "##templateId##",
  "EnvelopeOverrideOptions": {
    "Recipients": [
      {

        "OrderIndex": 1,
        "Email":"Placeholder:first signer",
        "Recipient": {
          "Email": "##email_of_the_signer##",
          "FirstName": "SomeName",
          "LastName": "SomeName",
          "LanguageCode": "en"
        }
        },
```

Copy Document from template

For this api call you just need the templateId. The response of this call is a **new SspFileId** and the workstep configuration of the template. You can use the SspFileId and the workstep configuration to send an envelope.

api endpoints:

- REST: /v4/envelope/{templateId}/copyFromTemplate
- SOAP: CopyDocumentFromTemplate_v1

Create envelope from template

You can also create an envelope from a template via api. You can find a configuration in the next section. After finishing the api call you will get a draftId for the created envelope.

```
{
```

```
"TemplateId": "##TemplateId##",
"EnvelopeOverrideOptions": {
  "Recipients": [
    {
      "RecipientId": "##RecipientId##",
      "OrderIndex": 1,
      "Email": "##EMAIL##",
      "Recipient": {
        "Email": "##EMAIL##",
        "FirstName": "##NAME##",
        "LastName": "##NAME##",
        "LanguageCode": "en",
        "EmailBodyExtra": "string",
        "DisableEmail": false,
        "AddAndroidAppLink": false,
        "AddIosAppLink": false,
        "AddWindowsAppLink": false,
        "AllowDelegation": true,
        "SkipExternalDataValidation": false,
        "AuthenticationMethods": [
          {
          }
        ],

        "SwissComCertificateData": {
          "PhoneNumber": "string",
          "MobileId": "string",
          "Parameters": [
            {
              "Key": "string",
              "Value": "string"
            }
          ]
        },
        "RemoteCertificateData": {
          "UserId": "string",
          "DeviceId": "string"
        },
        "OtpData": {
          "PhoneMobile": "string"
        },
        "Pkcs7SignerData": {
          "AllowedPkcs7SignatureTypes": [
            "LocalCertificate"
          ]
        }
      }
    }
  ],
  "AddFormFields": {
    "Forms": {


    }
  },
  "OverrideFormFieldValues": {
    "Forms": {


    }
  },
  "Name": "string",
  "EmailSubject": "string",
  "EmailBody": "string",
  "EnableReminders": true,
  "FirstReminderDayAmount": 1,
  "RecurrentReminderDayAmount": 1,
  "CallbackUrl": "string",
  "StatusUpdateCallbackUrl": "string",
  "WorkstepEventCallback": {
    "Url": "string",
```

```
      "Blacklist": [
        "string"
      ],
      "WhiteList": [
        "string"
      ]
    },
    "MetaDataXml": "string"
  },
  "CreateDraftOptions": {
    "AfterSendRedirectUrl": "string",
    "AfterSendCallbackUrl": "string",
    "RedirectPolicy": "None",
    "AllowAgentRedirect": true,
    "IframeWhiteList": "string"
  }
}
```

```
  <overrideOptions>
   <name>new name</name>
   <eMailSubject>new subject</eMailSubject>
   <eMailBody>new body</eMailBody>
   <callbackUrl></callbackUrl>
   <statusUpdateCallbackUrl></statusUpdateCallbackUrl>
   <enableReminders>true</enableReminders>
   <firstReminderDayAmount>1</firstReminderDayAmount>
   <recurrentReminderDayAmount>2</recurrentReminderDayAmount>
   <beforeExpirationReminderDayAmount>1</beforeExpirationReminderDayAmount>
   <daysUntilExpire>20</daysUntilExpire>
   <recipientOverrides>
      <recipientOverride>
         <recipientLookUp>
            <eMail>##EMAIL##</eMail>
            <orderIndex>1</orderIndex>
         </recipientLookUp>
         <recipient>
            <eMail>##EMAIL##</eMail>
            <emailBodyExtra>some extra text</emailBodyExtra>
            <firstName>Hello</firstName>
            <lastName>World</lastName>
            <languageCode>en</languageCode>
         </recipient>
      </recipientOverride>
   </recipientOverrides>
</overrideOptions>
```

For a SOAP implementation the following information (draftOptions) must be also filled in:

```
  <draftOptions>
    <afterSendRedirectUrl></afterSendRedirectUrl>
    <afterSendCallbackUrl></afterSendCallbackUrl>
    <redirectPolicy>ToDesigner</redirectPolicy>
</draftOptions>
```

Send envelope from template

Now you can send an envelope from a template. For this call you need the template Id as well as the overrideOptions. The next configurations show you the override options:

> The configuration has not all settings included. It is just for testing the API call with some simple configuration. The whole configuration you can find in the next collapse.

```
{
  "TemplateId": "##TemplateId##",
  "EnvelopeOverrideOptions": {
    "Recipients": [
      {
        "RecipientId": "##RecipientId##",
        "OrderIndex": 1,
        "Email": "##EMAIL##",
        "Recipient": {
          "Email": "##EMAIL##",
          "FirstName": "TESTNAME",
          "LastName": "TESTLASTNAME",
          "LanguageCode": "en",
          "EmailBodyExtra": "string",
          "OtpData": {
            "PhoneMobile": "string"
          },
          "Pkcs7SignerData": {
            "AllowedPkcs7SignatureTypes": [
              "LocalCertificate"
            ]
          }
        }
      }
    ],
    "AddFormFields": {
      "Forms": {

      }
    },
    "OverrideFormFieldValues": {
      "Forms": {

      }
    },
    "Name": "TEST",
    "EmailSubject": "TESTEMAILSUBJECT",
    "EmailBody": "string",
    "EnableReminders": true,
    "FirstReminderDayAmount": 2,
    "RecurrentReminderDayAmount": 1,
    "BeforeExpirationDayAmount": 1,
    "DaysUntilExpire": 25,
    "CallbackUrl": "string",
    "StatusUpdateCallbackUrl": "string",
    "MetaDataXml": "string"
  }
}
```

```
 <overrideOptions>

   <name>Sending with template</name>
   <eMailSubject>Subject of the envelope</eMailSubject>
   <eMailBody>Please sign the document!</eMailBody>
   <callbackUrl></callbackUrl>
   <statusUpdateCallbackUrl></statusUpdateCallbackUrl>
   <enableReminders>true</enableReminders>
   <firstReminderDayAmount>1</firstReminderDayAmount>
   <recurrentReminderDayAmount>2</recurrentReminderDayAmount>
   <beforeExpirationReminderDayAmount>1</beforeExpirationReminderDayAmount>
   <daysUntilExpire>20</daysUntilExpire>

   <recipientOverrides>
      <recipientOverride>
                 <recipientLookUp>
            <eMail>##EMAIL##</eMail>
            <orderIndex>1</orderIndex>
                 </recipientLookUp>
         <recipient>
            <eMail>##EMAIL##</eMail>
            <emailBodyExtra>Some extra text</emailBodyExtra>
            <firstName>FIRSTNAME</firstName>
            <lastName>LASTNAME</lastName>
            <languageCode>en</languageCode>

         </recipient>
      </recipientOverride>
   </recipientOverrides>
</overrideOptions>
```

The next configuration shows you a complete configuration in REST (JSON).

```
{
  "TemplateId": "string",
  "EnvelopeOverrideOptions": {
    "Recipients": [
      {
        "RecipientId": "string",
        "OrderIndex": 0,
        "Email": "string",
        "Recipient": {
          "Email": "string",
          "FirstName": "string",
          "LastName": "string",
          "LanguageCode": "string",
          "EmailBodyExtra": "string",
          "DisableEmail": false,
          "AddAndroidAppLink": false,
          "AddIosAppLink": false,
          "AddWindowsAppLink": false,
          "AllowDelegation": true,
          "SkipExternalDataValidation": false,
          "AuthenticationMethods": [
            {
              "Method": "Pin",
              "Parameter": "string",
              "Filters": [
                {
                  "CompareOperation": "Equals",
                  "FilterId": "string",
                  "FilterValue": "string"
                }
              ]
            }
          ],
          "DisposableCertificateData": {
            "CountryResidence": "string",
```

```
        "IdentificationCountry": "string",
        "IdentificationType": 1,
        "PhoneMobile": "string",
        "DocumentType": "CI",
        "DocumentIssuedBy": "string",
        "DocumentIssuedOn": "2019-10-21T07:07:19.513Z",
        "DocumentExpiryDate": "2019-10-21T07:07:19.513Z",
        "SerialNumber": "string",
        "DocumentNumber": "string",
        "OverrideHolderInCaseOfMismatch": true
      },
      "SwissComCertificateData": {
        "PhoneNumber": "string",
        "MobileId": "string",
        "Parameters": [
          {
            "Key": "string",
            "Value": "string"
          }
        ]
      },
      "RemoteCertificateData": {
        "UserId": "string",
        "DeviceId": "string"
      },
      "OtpData": {
        "PhoneMobile": "string"
      },
      "Pkcs7SignerData": {
        "AllowedPkcs7SignatureTypes": [
          "LocalCertificate"
        ]
      }
    }
  }
],
"AddFormFields": {
  "Forms": {
    "additionalProp1": [
      {
        "Name": "string",
        "ReadOnly": false,
        "Required": false,
        "X": 0,
        "Y": 0,
        "Width": 0,
        "Height": 0,
        "Page": 0,
        "DiscriminatorType": "string"
      }
    ],
    "additionalProp2": [
      {
        "Name": "string",
        "ReadOnly": false,
        "Required": false,
        "X": 0,
        "Y": 0,
        "Width": 0,
        "Height": 0,
        "Page": 0,
        "DiscriminatorType": "string"
      }
    ],
    "additionalProp3": [
      {
        "Name": "string",
        "ReadOnly": false,
        "Required": false,
        "X": 0,
        "Y": 0,
```

```
            "Width": 0,
            "Height": 0,
            "Page": 0,
            "DiscriminatorType": "string"
          }
        ]
      }
    },
    "OverrideFormFieldValues": {
      "Forms": {
        "additionalProp1": [
          {
            "Name": "string",
            "DiscriminatorType": "string"
          }
        ],
        "additionalProp2": [
          {
            "Name": "string",
            "DiscriminatorType": "string"
          }
        ],
        "additionalProp3": [
          {
            "Name": "string",
            "DiscriminatorType": "string"
          }
        ]
      }
    },
    "Name": "string",
    "EmailSubject": "string",
    "EmailBody": "string",
    "EnableReminders": true,
    "FirstReminderDayAmount": 0,
    "RecurrentReminderDayAmount": 0,
    "BeforeExpirationDayAmount": 0,
    "DaysUntilExpire": 0,
    "CallbackUrl": "string",
    "StatusUpdateCallbackUrl": "string",
    "WorkstepEventCallback": {
      "Url": "string",
      "Blacklist": [
        "string"
      ],
      "WhiteList": [
        "string"
      ]
    },
    "MetaDataXml": "string"
  }
}
```

Override form fields

In the configuration for sending an envelope from a template you can find the following section:

```
"OverrideFormFieldValues": {
      "Forms": {

      }
    },
```

There you can override form fields.

To override a form field for example a Checkbox or a Radiobutton the following steps are necessary:

- Create a template

- Get the envelope (/envelope/{##envelopeID##}
- Copy the name of the form field
- Override the form field

Override Checkbox

In this section you can find a description and a sample of how to override a given checkbox from the template which is checked and try to override the value that the checkbox is not checked.

With the get envelope api call you get the workstep configuration of the template. In this configuration you can find the form fields which are placed on the template. Please see the next sample:

```
"Documents": [
        {
            "PageSizesInPoints": [
                {
                    "Height": 792.0,
                    "Width": 612.0
                }
            ],
            "DocRefNumber": 1,
            "FileName": "Test.pdf",
            "FormFields": [
                {
                    "Name": "CheckBox_bfa536d5-a360-1924-8adf-c6918ce4f982",
                    "Values": [
                        {
                            "Key": "",
                            "Value": "true"
                        }
                    ]
                }
            ]
        }
    ],
```

In the section *“Documents”:[* you can find all form fields which are placed on the template. In this sample there is only one form field with the name: *CheckBox_bfa536d5-a360-1924-8adf-c6918ce4f982*

For the next step we need the send envelope from template configuration and the name of the form field which we want to override. In the next section you can find a sample where the checkbox gets overridden.

```
{
  "TemplateId": "##TEMPLATE##",
  "EnvelopeOverrideOptions": {
    "Recipients": [
      {
        "RecipientId": "##RECIPIENT##",
        "OrderIndex": 1,
        "Email": "##EMAIL##",
        "Recipient": {
          "Email": "##EMAIL##",
          "FirstName": "##NAME##",
          "LastName": "##NAME##",
          "LanguageCode": "en",
          "EmailBodyExtra": "string",


          "OtpData": {
            "PhoneMobile": "string"
          },
          "Pkcs7SignerData": {
            "AllowedPkcs7SignatureTypes": [
              "LocalCertificate"
            ]
          }
        }
      }
    ],
    "AddFormFields": {
      "Forms": {


      }
    },
     "OverrideFormFieldValues": {
      "Forms": {
        "1": [{
            "DiscriminatorType" : "FillCheckBoxFormDescriptor",
            "Name": "CheckBox_bfa536d5-a360-1924-8adf-c6918ce4f982",
            "IsChecked": "false"
        }

        ]

      }
     },
    "Name": "TEST",
    "EmailSubject": "##Subject##",
    "EmailBody": "string",
    "EnableReminders": true,
    "FirstReminderDayAmount": 2,
    "RecurrentReminderDayAmount": 1,
    "BeforeExpirationDayAmount": 1,
    "DaysUntilExpire": 25,
    "CallbackUrl": "string",
    "StatusUpdateCallbackUrl": "string",
    "MetaDataXml": "string"
  }
  }
```

```
<overrideOptions>
       <name>new name</name>
       <eMailSubject>new subject</eMailSubject>
       <eMailBody>new body</eMailBody>
       <callbackUrl></callbackUrl>
       <statusUpdateCallbackUrl></statusUpdateCallbackUrl>
       <enableReminders>true</enableReminders>
       <firstReminderDayAmount>1</firstReminderDayAmount>
       <recurrentReminderDayAmount>2</recurrentReminderDayAmount>
       <beforeExpirationReminderDayAmount>1</beforeExpirationReminderDayAmount>
       <daysUntilExpire>20</daysUntilExpire>
       <overrideFormFieldValues>
              <document docRef="1">
                     <checkBox name="CheckBox_bfa536d5-a360-1924-8adf-c6918ce4f982">
                            <isChecked>0</isChecked>
                     </checkBox>
              </document>
       </overrideFormFieldValues>
</overrideOptions>
```

The checkbox is now not checked because of the following line:

```
"IsChecked": "false"
```

Override Radiobutton

In this section you can find a description and a sample of how to override a given radiobutton from the template. In the following sample no button is selected in the given template.

With the get envelope api call you get the workstep configuration of the template. In this configuration you can find the form fields which are placed on the template. Please see the next sample:

```
"Documents": [
       {
           "PageSizesInPoints": [
               {
                   "Height": 792.0,
                   "Width": 612.0
               }
           ],
           "DocRefNumber": 1,
           "FileName": "Test.pdf",
           "FormFields": [
               {
                   "Name": "RadioButton_a497ee93-41ef-c2a0-6df0-a587a60be818",
                   "Values": []
               }
           ]
       }
   ],
```

In the section *"Documents":[* you can find all form fields which are placed on the template. In this sample there is only one form field with the name: RadioButton_a497ee93-41ef-c2a0-6df0-a587a60be818

For the next step we need the send envelope from template configuration and the name of the form field which we want to override. In the next section you can find a sample where the radiobutton gets overridden.

> In the configuration you can find the section "SelectedItem" with the following value: RadioButton_a497ee93-41ef-c2a0-6df0-a587a60be818#XyzmoIndexSeparator#1
>
> You need this part: #XyzmoIndexSeparator#1 to select which button in the group should be overwritten. In this example, #XyzmoIndexSeparator **#1** means that the second button in the group should be overwritten. Note: List starts with 0, therefore, **#1** is the second item in the group to be selected.

```
{
  "TemplateId": "##TemplateId##",
  "EnvelopeOverrideOptions": {
    "Recipients": [
      {

        "OrderIndex": 1,
        "Email":"##Email##",
        "Recipient": {
          "Email": "##Email##",
          "FirstName": "##Name##",
          "LastName": "##Name##",
          "LanguageCode": "en"
        }
        },
        {

        "OrderIndex": 2,
        "Email":"##Email##",
        "Recipient": {
          "Email": "##Email##",
          "FirstName": "##Name##",
          "LastName": "##Name##",
          "LanguageCode": "en"
        }
      }
    ],

    "OverrideFormFieldValues": {
      "Forms": {
        "1": [{
            "DiscriminatorType" : "FillRadioButtonFormDescriptor",
            "Name": "RadioButton_a497ee93-41ef-c2a0-6df0-a587a60be818",
            "SelectedItem": "RadioButton_a497ee93-41ef-c2a0-6df0-a587a60be818#XyzmoIndexSeparator#1"
        }

        ]

      }
     }

  }
  }
```

```
<overrideOptions>

        <name>new name</name>
        <eMailSubject>new subject</eMailSubject>
        <eMailBody>new body</eMailBody>
        <callbackUrl></callbackUrl>
        <statusUpdateCallbackUrl></statusUpdateCallbackUrl>
        <enableReminders>true</enableReminders>
        <firstReminderDayAmount>1</firstReminderDayAmount>
        <recurrentReminderDayAmount>2</recurrentReminderDayAmount>
        <beforeExpirationReminderDayAmount>1</beforeExpirationReminderDayAmount>
        <daysUntilExpire>20</daysUntilExpire>
        <overrideFormFieldValues>
      <document docRef="1">
         <radioButtonGroup name="RadioButton_a497ee93-41ef-c2a0-6df0-a587a60be818">
            <selectedItemId>RadioButton_a497ee93-41ef-c2a0-6df0-a587a60be818#XyzmoIndexSeparator#1<
/selectedItemId>
         </radioButtonGroup>
      </document>
   </overrideFormFieldValues>
</overrideOptions>
```

For the override method the following DiscriminatorTypes are available:

- FillCheckBoxFormDescriptor
- FillComboBoxFormDescriptor
- FillListBoxFormDescriptor
- FillRadioButtonFormDescriptor
- FillTextBoxFormDescriptor