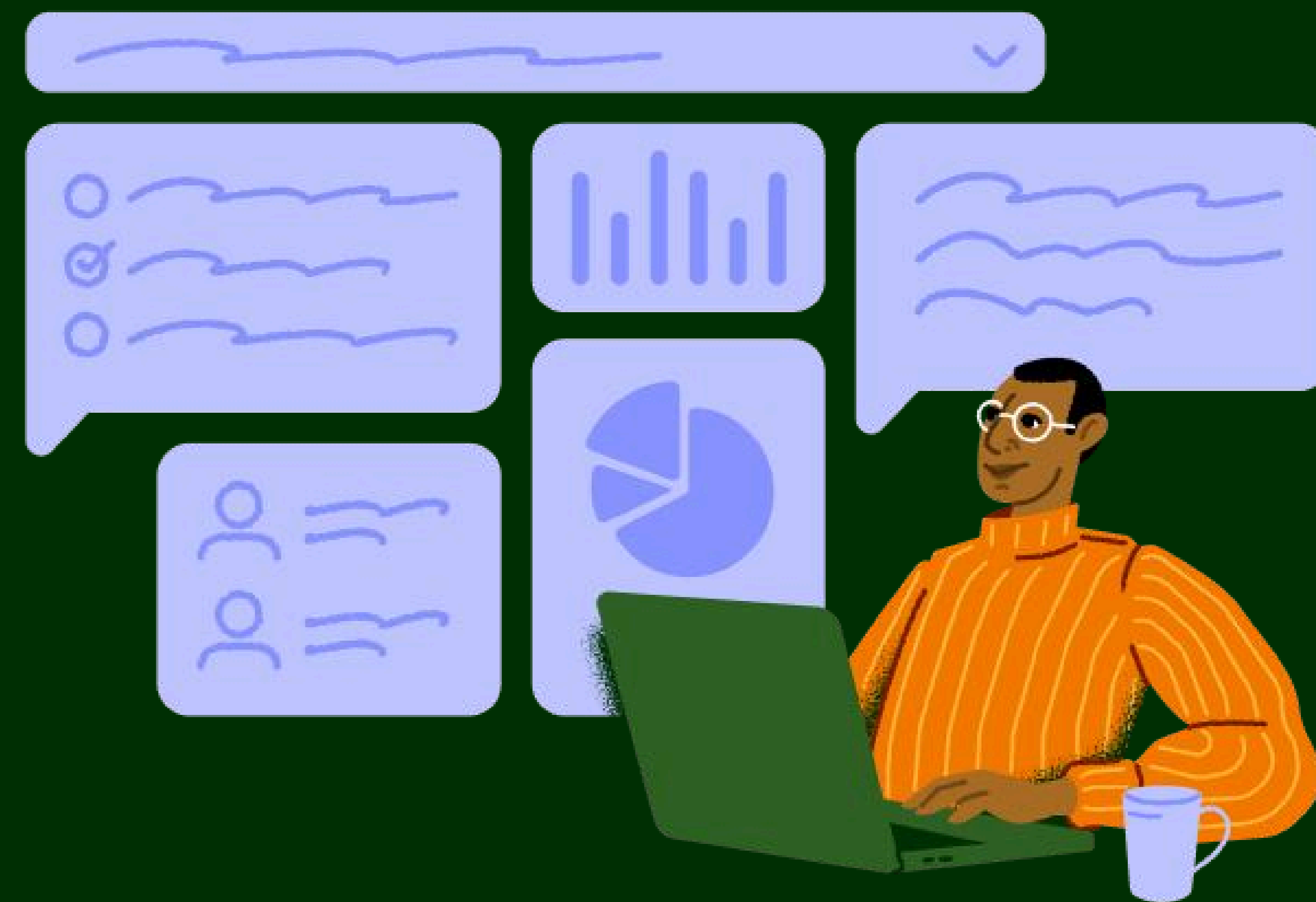


## Form Builder

The Form Builder is a visual tool for creating and testing forms. It generates a JSON structure that is used to create forms via Kivra's API as a next step. The tool does not send any data to Kivra, and we recommend testing everything in a sandbox before production.



### How It Works

The interface is divided into two main working modes: **Visual Mode** and **JSON Mode**. In Visual Mode, you build the form step by step, while JSON Mode shows the technical representation of the form.

On the right-hand side of the tool, a preview is always displayed. There you can test the form exactly as an end user would experience it live in Kivra, making it easy to verify both content and flow.

### Creating a Form

Start in Visual Mode by entering a title for the form, which is required. Then build the form by adding questions.

When adding a question, you must choose a question type and provide a title. Without a title, you cannot continue building the form. Questions are displayed in the order they are created, and unless specified otherwise, the form will automatically proceed to the next question in the list.

After creating the questions, you can use the preview on the right to test the form. This is an important step to ensure that both content and flow work as intended.

### Working in JSON Mode

When you switch to JSON Mode, the form is displayed in a JSON format. Here you can make more advanced adjustments that are not available in the visual mode, such as adding descriptive text.

JSON Mode is also used to understand how the form is technically represented, which is relevant when integrating with the API.

### Question Types

The tool supports several different question types depending on the type of response you want to collect.

**Free text** is used when the user should be able to write a free response without any restrictions.

**Number** is used when the answer should be numeric, and you can also specify minimum and maximum allowed values.

**Options** is used when the user should select one option. While **Multi Options**, is used when multiple selections should be allowed. If many options are added, they will be displayed as a dropdown to improve the user experience.

**Verified Account** allows the user to enter a bank account via their bank. In the tool, this flow is simulated and not connected to BankID or real banking services.

## Controlling the Flow of the Form

Each question in the form needs to point to the next step. If no choice is made, the next question is automatically set based on the order in the form.

For the Options question type, it is possible to control the flow depending on the user's answer. This allows different users to be guided through different parts of the form.

It is important that there is at least one question that does not point to another question, as this marks the end of the form.

The tool does not check for loops in the flow. This means that in some cases a user may get stuck in the form, making it especially important to test the entire flow before use.

## Validation

The form is validated directly in the tool. For the form to be valid, it must contain a title, a description, and at least one question.

There must be a starting question that points to an existing question in the form. All questions and answer options must have unique IDs, and all questions must be reachable within the flow.

Additionally, there must be at least one endpoint in the form. If any of these requirements are not met, an error message will be displayed.

## Saving and Sharing the Form

Once the form is complete, it can be saved either as a JSON file or as a URL. A downloaded file can be used for further development, while a URL makes it easy to save or share the form.



It is important to remember that the tool does not have a draft-saving function. If the browser is closed, changes may be lost.

## From Template to Production

When the form is complete, the JSON structure is exported and sent to a partner or integration. From there, Kivra's Forms API is called, which returns a unique `form_template_id`.

This ID is then used to create a dispatch via the Content API. When the recipient opens the form in Kivra, it can be filled in and submitted.

Each submitted response generates a unique key that is used to retrieve the answers via the API and deliver them back to the sender.