# 

# Creating Data Views in Kibana

Alright, just one thing left to do before we can really get started; to create data views. But first, let’s talk about what a data view is in the first place. Kibana retrieves the data it displays from Elasticsearch indices, whether it’s for visualizations, dashboards, or queries. That’s all done by using Elasticsearch’s REST API. Nothing new here.



## What is a Data View?

A data view specifies an index pattern, which matches one or more indices. Let me illustrate how data views work in the context of our test data. We have an index named 'orders' which stores all of our order data. Now we want to create a number of Kibana visualizations for this index. This could be a line or bar chart showing the number of orders over time, and a pie chart showing the most popular products. For each of the visualizations, we need to tell Kibana from which Elasticsearch indices it should retrieve the data. This is exactly what a data view does.

## Examples of Data Views

Let’s look at two examples of data views for two different use cases.

Our orders data set was indexed into a single index named 'orders'. When we create a data view for this data set, we can therefore simply enter that as the index pattern. Although we only have a single index containing orders, we might as well add a wildcard at the end, just in case we need to split the orders data into multiple indices in the future. This causes the data view to match data in all Elasticsearch indices that begin with 'orders'. If we change our indexing strategy in the future, we won’t need to modify our Kibana data view to reflect this; any visualizations or dashboards associated with this data view will automatically retrieve data from all matching indices.

This would be the case for our access logs data set. Specifically, we have three indices storing the access logs — one index per month. That’s because we are running a busy website, and so storing all of the access logs within a single index does not scale well enough. This approach is quite common with the Elastic Stack. As long as the index names use some kind of naming scheme, we can define an index pattern with a wildcard that matches all of them. In this case, we have the year and month number at the end of the index names, so that’s the part we need to match with a wildcard. With the data view that you see on your screen now, we can match our three indices and any that might be added later.

## Benefits of Using Wildcards in Index Patterns

Using wildcards in index patterns is a way of dealing with data that is divided into multiple indices. This enables us to work with the data as if it came from a single index, even though several indices are queried within Elasticsearch. Pretty cool, right?

## Creating Data Views

A data view is actually required before we can interact with our data within Kibana, so let’s create one. We will need to create a data view for each of our two datasets.

Let’s begin with the access logs dataset. First, click the 'Stack Management' menu item within the 'Management' group. Then find the 'Kibana' group in the secondary menu, under which we click 'Data Views'. Then click the 'Create data view' button to get started.

I'll begin by creating a data view for our access logs, so I will name it 'Access logs'. Next up, we need to define an index pattern. I will just use the one I showed you on the diagram a moment ago. As you can see to the right, this pattern matches all of our three indices. To make using our data view smooth and easy, we can configure a timestamp field. Kibana will then use this field automatically for filtering our data by time. The correct field has already been chosen for us, so we don’t need to do anything. That’s all we need to do, so let’s create the data view.

## Finalizing Data Views

The data view has now been created. Kibana has inspected the field mappings of the matching indices, which you can see within the table. Let’s finish up by creating a data view for our orders dataset as well. Let’s name this one 'Orders'. Even though we only have one index for this dataset, I will still include a wildcard in the index pattern. That’s it! We now have a data view for each of our datasets. Everything is now set up, and we are ready to get started with Kibana.