# KIP-569: DescribeConfigsResponse - Update the schema to include additional metadata information of the field

## Status

**Current state**: Accepted

**Discussion thread**: here

**JIRA**: ⚠ Unable to render Jira issues macro, execution error.

**PR**:

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

## Motivation

The admin client allows users to describe the configuration of brokers and topics in the cluster via `describeConfigs` method. The returned configuration includes name, default values, source, configuration synonym including if the information is sensitive `isSensitive`, or is read only `isReadOnly`. The consumer of this API can choose to display this information in GUI e.g. web forms in Kafka lenses, Confluent Control Center. While displaying this information, fields such as `isSensitive` and `isReadOnly` can be used by the client to hide sensitive information and to prevent the user from editing the field respectively. This provides better upfront experience to the user.

This KIP proposes to include 2 additional metadata information - `dataType`, and `documentation` for the individual config entries.

The motivation behind including `dataType` is to enable the client to provide better validation to the user. In the absence of `dataType` the only way to know if the user specified value is correct is to make an `alter` call and check if there is no error, which is a sub-optimal experience.  Including the `dataType` could open up potentially other use cases as well.

Including `documentation` would enable the client to provide latest information about the config and would also act as a single source of truth.

## Public Interfaces

### Network protocol

#### Request

The changes to the Request schema include

1. Bump up the valid versions range
2. New field `IncludeDocumentation` - with default value of false

```
{
  "apiKey": 32,
  "type": "request",
  "name": "DescribeConfigsRequest",
  // Version 1 adds IncludeSynoyms.
  // Version 2 is the same as version 1.
  // Version 3 is the same as version 2.
  "validVersions": "0-3", <--- Updated Field
  "flexibleVersions": "none",
  "fields": [
    { "name": "Resources", "type": "[]DescribeConfigsResource", "versions": "0+",
      "about": "The resources whose configurations we want to describe.", "fields": [
      { "name": "ResourceType", "type": "int8", "versions": "0+",
        "about": "The resource type." },
      { "name": "ResourceName", "type": "string", "versions": "0+",
        "about": "The resource name." },
      { "name": "ConfigurationKeys", "type": "[]string", "versions": "0+", "nullableVersions": "0+",
        "about": "The configuration keys to list, or null to list all configuration keys." }
    ]},
    { "name": "IncludeSynoyms", "type": "bool", "versions": "1+", "default": "false", "ignorable": false,
      "about": "True if we should include all synonyms." }
        { "name": "IncludeDocumentation", "type": "bool", "versions": "1+", "default": "false", "ignorable":
false,  <--- New Field
      "about": "True if we should include configuration documentation." }
  ]
}
```

## Response

The changes to the Response schema include

1. Bump valid versions range
2. Add new Fields - `ConfigValueType and Documentation`

```
{
  "apiKey": 32,
  "type": "response",
  "name": "DescribeConfigsResponse",
  "validVersions": "0-3", <--- Updated Field
  "flexibleVersions": "none",
  "fields": [
    { "name": "ThrottleTimeMs", "type": "int32", "versions": "0+",
      "about": "The duration in milliseconds for which the request was throttled due to a quota violation, or
zero if the request did not violate any quota." },
    { "name": "Results", "type": "[]DescribeConfigsResult", "versions": "0+",
      "about": "The results for each resource.", "fields": [
      { "name": "ErrorCode", "type": "int16", "versions": "0+",
        "about": "The error code, or 0 if we were able to successfully describe the configurations." },
      { "name": "ErrorMessage", "type": "string", "versions": "0+", "nullableVersions": "0+",
        "about": "The error message, or null if we were able to successfully describe the configurations." },
      { "name": "ResourceType", "type": "int8", "versions": "0+",
        "about": "The resource type." },
      { "name": "ResourceName", "type": "string", "versions": "0+",
        "about": "The resource name." },
      { "name": "Configs", "type": "[]DescribeConfigsResourceResult", "versions": "0+",
        "about": "Each listed configuration.", "fields": [
        { "name": "Name", "type": "string", "versions": "0+",
          "about": "The configuration name." },
        { "name": "Value", "type": "string", "versions": "0+", "nullableVersions": "0+",
          "about": "The configuration value." },
        { "name": "ReadOnly", "type": "bool", "versions": "0+",
          "about": "True if the configuration is read-only." },
        { "name": "IsDefault", "type": "bool", "versions": "0",
          "about": "True if the configuration is not set." },
        // Note: the v0 default for this field that should be exposed to callers is
        // context-dependent. For example, if the resource is a broker, this should default to 4.
        // -1 is just a placeholder value.
        { "name": "ConfigSource", "type": "int8", "versions": "1+", "default": "-1", "ignorable": true,
          "about": "The configuration source." },
        { "name": "IsSensitive", "type": "bool", "versions": "0+",
          "about": "True if this configuration is sensitive." },
        { "name": "Synonyms", "type": "[]DescribeConfigsSynonym", "versions": "1+", "ignorable": true,
          "about": "The synonyms for this configuration key.", "fields": [
          { "name": "Name", "type": "string", "versions": "1+",
            "about": "The synonym name." },
          { "name": "Value", "type": "string", "versions": "1+", "nullableVersions": "0+",
            "about": "The synonym value." },
          { "name": "Source", "type": "int8", "versions": "1+",
            "about": "The synonym source." }
        ]},
        { "name": "ConfigValueType", "type": "int8", "versions": "3+", default": "0",     <--- New Field
          "about": "The configuration data type."},
        { "name": "Documentation", "type": "string", "versions": "3+", "nullableVersions": "0+",    <--- New
Field
          "about": "The configuration documentation." },
      ]}
    ]}
  ]
}
```

## AdminClient Class

No new method will be added or updated to the `AdminCient` class.

## AbstractConfig Class

As part of this KIP, a new method will be added to AbstractConfig to return the documentation of the given Config

```
public String documentationOf(String key) {
        ConfigDef.ConfigKey configKey = definition.configKeys().get(key);
        if (configKey == null)
            return null;
        return configKey.documentation;
    }
```

# Proposed Changes

Under the proposed change, response for `AdminClient.describeConfigs` would include following new properties, which would be part of *`DescribeConfigsResponse.ConfigEntry`* class.

1. `Type: ConfigType` - This is an Enum of all the supported data type.

   ```
   /**
    * Data type of configuration entry.
   */
   public enum ConfigType {
      UNKNOWN,
      BOOLEAN,
      STRING,
      INT,
      SHORT,
      LONG,
      DOUBLE,
      LIST,
      CLASS,
      PASSWORD
   }
   ```

   `ConfigType.Unknown` is for ensuring backward compatibility (explained is Compatibility section below)

   The values in above Enum is derived from existing `Type` Enum defined in ConfigDef.java below

   ```
    /**
    * The config types
    */
    public enum Type {
      BOOLEAN, STRING, INT, SHORT, LONG, DOUBLE, LIST, CLASS, PASSWORD
    }
   ```

2. `Documentation: String` - field's documentation

`DescribeConfigsResponse.ConfigEntry` with the above new fields

```
public class ConfigEntry {
    private final String name;
    private final String value;
    private final ConfigSource source;
    private final boolean isSensitive;
    private final boolean isReadOnly;
    private final List<ConfigSynonym> synonyms;

.....
    private final ConfigType type; // This is the new property
        private final String documentation; // This is the new property
.....
```

Rest of the changes would be plumbing this new property in `AdminManager` class to be eventually consumed by `AdminClient` class

## Sample Response

Following example show how the response would look like in JSON format.
*Note: Kafka uses binary protocol over TCP. ConfigType Enum is serialized as byte instead of string.*

```
{
  "name": "offsets.topic.num.partitions",
  "value": "50",
  "source": "DEFAULT_CONFIG",
  "type": "INT",   // <---- New Field
  "synonyms": [
    {
      "name": "offsets.topic.num.partitions",
      "value": "50",
      "source": "DEFAULT_CONFIG"
    }
  ],
  "documentation": "The number of partitions for the offset commit topic (should not change after deployment)",
//  <---- New Field
  "isReadOnly": true,
  "isSensitive": false,
 }
```

## Default Behavior

By default, the `documentation` field will not be included in the response to `describeConfigs`. This is to avoid the response from boating up as the value for `documentation` field can be quite large.

Below example shows how client can include the new fields in the response

**Example**

```
KafkaAdminClient.describeConfigs(
  resources,
  new DescribeConfigsOptions()
       .includeSynonyms(true)
       .includeDocumentation(true)
)
```

# Compatibility, Deprecation, and Migration Plan

For backward compatibility we rely on existing schema versioning technique. Under this, client sends its version information in the request header. This in turn is used on the server to select the schema for that version.

| | Server version | Client version | Expected Behavior |
|---|---|---|---|
| 1 | X | X | Both `Type` and `Documentation` is visible to the client. |
| 2 | X | X-1 | `Type` and `Documentation` are not visible to the client.<br>Not serialized as part of the response. |
| 3 | X-1 | X | IllegalArgumentException: Invalid version for API key... |
| 4 | X+1 | X | Both `Type` and `Documentation` is visible to the client.<br>If version X+1 has new value in enum `ConfigType`, client would get `ConfigType.Unknown` instead for that new value. |

# Rejected Alternatives

*NA*