

Strongly typed fields

At its core, JavaScript is a loosely typed language, which means you do not have to declare the data types of variables explicitly. JavaScript variables assume a data type based on the content that is placed in them. In many cases, JavaScript performs conversions automatically when they are needed. For example, if you add a number to an item that consists of text (a string), the number is converted to text automatically.

However, for business applications, being able to explicitly specify a data type is extremely useful. For example, you may want to strongly type an amount field, so that it can only hold numeric data with a specific precision (length and decimal positions). In a business application, putting character data into an amount field should raise an exception. It is for this reason that Profound.js introduces strong types to JavaScript. You can now use both loosely typed variables and strongly typed fields within your applications. Profound.js supports a variety of data types, including all native IBM i data types, pointers, fixed length character fields, and data structures. In addition to making your application code more robust, this capability also allows for simple communication to take place between your Node.js applications and various IBM i resources, such as system API and programs.

A strongly typed field is declared using the [pjs.define\(\)](#) API.

The following examples illustrate the difference between loosely and strongly typed fields.

Loosely typed variable

```
var amount;           // amount is a loosely typed variable and has not been initialized
amount = 125.99;     // amount is now a number
amount = "ABC";      // this is valid because amount is loosely typed; it is converted to a string
on the fly
```

Strongly typed field

```
pjs.define("amount", { type: "packed decimal", length: 7, decimals: 2 }); // amount is strongly
typed and initialized to 0
amount = 125.99; // amount has changed from 0 to 125.99
amount = "ABC"; // this line of code will produce an error because amount can only hold
numbers with a predefined precision
```