

IZABELA SZCZEGODZIŃSKA

EMIL ZYGALA

MICHAŁ KULBIŃSKI

PROJEKT "WALUTY"



- Naszym celem było stworzenie aplikacji dostarczającej podstawowych informacji na temat rynku walutowego
- Chcieliśmy stworzyć narzędzia umożliwiające szybkie porównywanie cen wybranych walut zarówno w zakresie wartości historycznych jak i cen bieżących

MOTYWACJA I CEL PROJEKTU

- Dane, z których korzysta nasza aplikacja są dostarczane przez NBP i pochodzą ze strony <http://api.nbp.pl/en.html>
- Korzystamy z tabel kursów średnich A i B, które zawierają łącznie kursy 150 walut w przeliczeniu na PLN

WYKORZYSTANE DANE

Kierowani chęcią stworzenia aplikacji dostarczającej w miarę kompleksowych informacji o cenach walut zdecydowaliśmy się na wprowadzenie następujących funkcjonalności:

- Zestawianie w postaci wykresu aktualnych cen walut wyrażonych w PLN
- Tworzenie wykresów pozwalających na porównanie kształtowania się cen w czasie
- Kalkulator walut
- Ostrzeżenia na temat gwałtownych zmian cen

WPROWADZONE FUNKCJONALNOŚCI

```
158
159 actual_prices <- function(waluty) {
160
161     path <- "http://api.nbp.pl/api/exchangerates/tables/a/"
162     df <- ldpoly(fromJSON(path), data.frame)
163     df = df[4:length(df)]
164     iter = length(df)/3
165     waluta = c()
166     wartosc = c()
167     for (i in 1:iter) {
168         if (df[3*i - 1] %in% waluty) {
169             waluta <- c(waluta, df[3*i - 1])
170             wartosc <- c(wartosc, df[3*i])
171         }
172     }
173
174     dane <- data.frame("waluta" = as.character(c(waluta)),
175                       "wartosc" = as.double(c(wartosc)))
176 }
177
```

ZESTAWIENIE AKTUALNYCH CEN

```

41 cur_rate <- function(cur, startDate, endDate, names_a, names_b) {
42
43   if(cur %in% names_a[,1])
44     path <- "http://api.nbp.pl/api/exchangerates/rates/a/code/startDate/endDate/"
45
46   else if(cur %in% names_b[,1])
47     path <- "http://api.nbp.pl/api/exchangerates/rates/b/code/startDate/endDate/"
48
49   startDate <- as.Date(startDate)
50   endDate <- as.Date(endDate)
51   days <- as.integer(difftime(endDate,startDate))
52
53
54   if(days < 90){
55     path <- gsub("startDate",startDate,path)
56     path <- gsub("endDate",endDate,path)
57     path <- gsub("code",cur,path)
58     data <- ldply(fromJSON(path), data.frame)
59     row <- which(data == "rates")
60     data <- data.table(
61       unname(unlist(data[row,grep("mid",names(data))])),
62       unname(unlist(data[row,grep("Date",names(data))]))
63     )
64     colnames(data) <- c("rate","date")
65
66     data <- data[,c(2,1)]
67     data <- cbind(data,rep(cur,nrow(data)))
68     colnames(data)[3] <- "sign"
69     return(data)
70   }
71
72   iter <- ceiling(days/91)
73   intervals <- c(seq(0,days,91),days)
74

```

```

74   for(i in 1:(length(intervals)-1)){
75
76
77     path_c <- gsub("startDate",startDate+intervals[i],path)
78     path_c <- gsub("endDate",startDate+intervals[i+1],path_c)
79     path_c <- gsub("code",cur,path_c)
80     print(path_c)
81     if(i == 1){
82
83       data <- ldply(fromJSON(path_c), data.frame)
84       row <- which(data=="rates")
85       data <- data.table(
86         unname(unlist(data[row,grep("mid",names(data))])),
87         unname(unlist(data[row,grep("Date",names(data))]))
88       )
89       colnames(data) <- c("rate","date")
90     }
91     else{
92       x <- ldply(fromJSON(path_c), data.frame)
93       row <- which(x=="rates")
94       x <- data.table(
95         unname(unlist(x[row,grep("mid",names(x))])),
96         unname(unlist(x[row,grep("Date",names(x))]))
97       )
98       data <- rbind(data,x,use.names=FALSE)
99     }
100   }
101   colnames(data) <- c("rate","date")
102   data <- unique(data[,c(2,1)])
103   data <- cbind(data,rep(cur,nrow(data)))
104   colnames(data)[3] <- "sign"
105   return(data)
106 }
107

```

PORÓWNYWANIE CEN W CZASIE

```

107
108 compar_cur <- function(cur, startDate, endDate) {
109
110   for(c in cur){
111     if(which(cur==c)[1]==1){
112       data <- cur_rate(c, startDate, endDate, names_a, names_b)
113
114       colnames(data) <- c("date","rate","sign")
115     }
116     else{
117       new <- cur_rate(c, startDate, endDate, names_a, names_b)
118       colnames(new) <- c("date","rate","sign")
119       data <- rbind(data,new)
120     }
121   }
122
123   data$date = as.Date(data$date, origin = "1964-10-22")
124   return(data)
125 }
126

```

```

127
128 move_of_cur <- function(cur, names_a, names_b) {
129   date <- Sys.Date()
130   a <- cur[cur %in% names_a[,1]]
131   b <- cur[cur %in% names_b[,1]]
132
133   if(length(a) != 0){
134
135     data_a <- compar_cur(a, date-5, date)
136     x <- data_a$date
137     x <- max( x[x!=max(x)] )
138     data_a <- data_a[data_a$date >= x,]
139     data_a <- spread(data_a,key = date, value = rate)
140     data_a <- cbind(data_a, 0 < data_a[,3]-data_a[,2])
141     colnames(data_a)[4] <- "is_increasing"
142   }
143
144   if(length(b) != 0){
145     data_b <- compar_cur(b, date-15, date)
146     x <- data_b$date
147     x <- max( x[x!=max(x)] )
148     data_b <- data_b[data_b$date >= x,]
149     data_b <- spread(data_b,key = date, value = rate)
150     data_b <- cbind(data_b, 0 < data_b[,3]-data_b[,2])
151     colnames(data_b)[4] <- "is_increasing"
152   }
153   if(length(a) == 0) return(data_b[,c(1,4)])
154   if(length(b) == 0) return(data_a[,c(1,4)])
155
156   return(rbind(data_a,data_b,use.names= FALSE)[,c(1,4)])
157 }
158

```

ALERTY O GWAŁTOWNYCH ZMIANACH


```
178
179 value_in_pln <- function(symb, date="") {
180   names_a = names_by_table_a()
181   names_b = names_by_table_b()
182   if(symb %in% as.vector(names_a[,1]))
183     path <- "http://api.nbp.pl/api/exchangerates/rates/a/code/date"
184   else if(symb %in% names_b[,1])
185     path <- "http://api.nbp.pl/api/exchangerates/rates/b/code/date"
186   path <- gsub("code", symb, path)
187   path <- gsub("date", date, path)
188   result <- ldply(fromJSON(path), data.frame)[4,5]
189   return(result)
190 }
191
192
193 converter <- function(nr_units, symb1, symb2, date) {
194   nr_units = as.double(nr_units)
195   result <- nr_units*value_in_pln(symb1, date)/value_in_pln(symb2, date)
196   return(round(result, 2))
197 }
```

KALKULATOR WALUT

**DZIĘKUJEMY ZA
UWAGĘ**