

# Оквирна спецификација апликативног интерфејса за непосредан приступ систему електронских фактура

## Увод

Чланом 7. Правилника о начину и поступку регистравања за приступ систему електронских фактура, начину приступања и коришћења система електронских фактура и начину коришћења података који су доступни у систему електронских фактура („Службени гласник РС”, број 69/21, у даљем тексту: Правилник) предвиђено је да за непосредан приступ систему електронских фактура постоје омогућена два техничка решења:

- 1) путем корисничког интерфејса и када овлашћено лице путем интернет претраживача може да врши креирање и издавање електронских фактура и/или прихватање и одбијање електронских фактура и електронско евидентирање обрачуна пореза на додату вредност у име субјекта промета у оквиру својих овлашћења;
- 2) путем апликативног интерфејса (енг. *API – Application Programming Interface*) који омогућава повезивање са софтверским системом субјекта промета и вршење истих радњи као из тачке 1).

Овај документ даје оквирну спецификацију за приступ путем апликативног интерфејса.

Крајња техничка спецификација апликативног интерфејса је саставни део имплементације апликативног интерфејса и одржава се тако да увек буде усаглашена са одговарајућом верзијом имплементираних система.

У цитираној одредби Правилника се наводи да апликативни интерфејс омогућава повезивање са софтверским системом субјекта промета, за шта ћемо у даљем тексту користити ознаку *ERP* (од енглеског назива *enterprise resource planner*).

Основни сценарио који треба да буде подржан кроз апликативни интерфејс је да субјект који користи систем електронских фактура има свој *ERP*, запослени који су укључени у процесе издавања фактура и обраде примљених фактура све активности спроводе кроз *ERP*, а *ERP* аутоматски у позадини преко апликативног интерфејса:

- иницира издавање фактура кроз систем електронских фактура (тј. њихово слање),
- преузима издате фактуре,
- преузима примењене фактуре,
- шаље информацију о прихватању или одбијању фактуре,
- преузима пристигле информације о прихватању или одбијању издатих фактура,
- врши електронско евидентирање обрачуна ПДВ-а.

Претпоставља се да приступ систему електронских фактура не мора бити на доступан у континуитету 24 сата дневно, већ на постоји могућност да неколико сати у току ноћи систем не буде доступан за кориснике. Користићемо термин ноћна пауза за период у коме није предвиђено да систем електронских фактура буде доступан.

Сличну ноћну паузу имају многи системи, на пример електронски сервис на порталу "еПорези" нису доступни у периоду од поноћи до 6 часова ујутро.

Апликативни интерфејс обухвата следеће операције:

- 1) иницирање издавања (тј. слања) електронске фактуре – операција одмах враћа идентификатор фактуре не чекајући да се заврши трансакција издавања фактуре у систему електронских фактура
- 2) статус појединачне издате фактуре
- 3) садржај појединачне издате фактуре
- 4) статус појединачне примљене фактуре
- 5) садржај појединачне примљене фактуре
- 6) регистрација *callback* операције за нотификације о издатим фактурама, примљеним фактурама и променама статуса издатих и примљених фактура
- 7) упит промена за примљене фактуре - преузимање листе идентификатора за фактуре примљене или са промењеним статусом на одређен датум, искључујући текући датум
- 8) упит промена за издате фактуре - преузимање листе идентификатора за фактуре издате или са промењеним статусом на одређен датум, искључујући текући датум
- 9) исказивање прихватања или одбијања примљене фактуре
- 10) операције за електронско евидентирање обрачуна ПДВ

*ERP* не мора регистровати *callback* операцију и имати имплементиран сервис који експонира ту операцију, али у том случају *ERP* не преузима промене које настају текућег дана. То је слично као када се из платног промета књиже само дневни изводи банке, али не и промене настале у току текућег дана. Чак и ако *ERP* не преузима промене за текући дан и даље је могуће видети фактуре пристигле текућег дана кроз кориснички интерфејс (као што је у апликацији за е-банкарство могуће видети и уплате које још увек нису прокњижене у *ERP*).

### Подаци о електронској фактури

Разликоваћемо три групе података о електронској фактури:

- 1) основни садржај фактуре – подаци садржани у UBL формату фактуре
- 2) додатни подаци фактуре – подаци који се придружују фактури приликом издавања у систему електронских фактура и не уписују се у UBL фактуре, као што је идентификатор фактуре
- 3) статусни подаци – подаци који се придружују или се могу мењати након издавања фактуре, као што је информација о прихватању или одбијању фактуре

Када се из система електронских фактура преузима фактура са придруженим додатним подацима онда се користи XML формат који додаје омот на UBL формат фактуре. Тачна спецификација формата омота је део крајње техничке спецификације апликативног интерфејса, а као илустрацију дајемо могућу структуру:

```
<env:DocumentEnvelope xmlns:env="...">
  <env:DocumentHeader>
    <env:DocumentId>6946581a-99b0-470c-ad60-d60d8fdaf9c5</env:DocumentId>
    ...
  </env:DocumentHeader>
  <env:DocumentBody>
    <Invoice ...>
      ...
    </Invoice>
  </env:DocumentBody>
</env:DocumentEnvelope>
```

## Аутентикација

Чланом 4. Правилника је предвиђено да је у апликацији за регистрацију могуће креирати податке за аутентикацију путем апликативног интерфејса.

Прецизан опис начина на који ће се користити подаци за аутентикацију је део крајње техничке спецификације апликативног интерфејса и може предвидети додатне параметре операција у односу на минимално одређене овом оквирном спецификацијом.

У сваком случају подаци за аутентикацију у себи садрже идентификатор клијентске стране.

## Поуздан пренос података

Сваки позив операције апликативног интерфејса може да се заврши на три начина:

- успешно
- уредно неуспешно
- неуредно

Уредно неуспешно значи да је друга страна одговорила и у одговору стоји да из одређеног разлога позвана операција није успешно извршена. Тада можемо рачунати да операција заиста није успела.

Неуредно значи да је дошло до грешке на начин да нисмо добили уредан одговор од друге стране, па макар тај одговор био да операција није успела. Код неуредно завршеног позива постоји могућност да је операција успешно извршена и произвела одговарајуће последице, али да одговор на позив није стигао назад због специфичних техничких проблема.

Да би обезбедили поуздан пренос података у коме неће долазити ни до губитака ни до дуплирања, потребно је решити сценарио када операција треба да се изврши тачно једном (као што је издавање одређене фактуре), а позив операције је завршен неуредно. Изазов у том сценарију је питање да ли треба поновити позив.

Решење за позиве операција које су осетљиве на дуплирање је:

- као додатни параметар операције се уводи јединствени идентификатор захтева који одређује страна која иницира позив при сваком позиву
- ако се позив заврши неуредно онда страна која иницира позив понавља позив са истим идентификатором захтева, а иначе се исти идентификатор захтева више не користи
- страна која прима позив проверава да ли је раније већ обрађен позив са истим идентификатором захтева и:
  - ако јесте, враћа се исти одговор који је враћен или је требало да буде враћен приликом првобитног позива,
  - ако није, онда се обради позив и у оквиру обраде се сачува довољно информација за случај понављања позива са истим идентификатором захтева.

## Операције апликативног интерфејса

Апликативни интерфејс је типа *REST API*. То, између осталог, значи да је свака операција одређена методом *HTTP* захтева (*method*) и путањом (*path*) која је део *URL*-а. Када се у даљем тексту приликом навођења путање неки њен део наведе у витичастим заградама, то значи да тај део представља параметар, а оно што је у витичастим заградама је назив параметра.

Коначни апликативни интерфејс може имати додатне операције, као и додатне параметре и варијанте за исту операцију.

#### Иницирање издавања електронске фактуре

*method:* POST

*path:* /api/publicApi/sales-invoice/ubl/upload/{requestId}

Минималан скуп параметара је:

- јединствени идентификатор захтева (параметар *requestId*)
- основни садржај фактуре у UBL формату (у телу *HTTP* захтева)

Јединствени идентификатор захтева треба користити како је описано у одељку о поузданом преносу података.

У одговору се, уколико је успешан, у *JSON* речнику налази:

- идентификатор фактуре (кључ *salesInvoiceId*)

Успешан одговор и даље не мора да значи да је фактура успешно издата јер ова операција не чека да се заврши обрада издавања фактуре.

#### Статус појединачне издате фактуре

*method:* GET

*path:* /api/publicApi/sales-invoice/{invoiceId}

Минималан скуп параметара је:

- идентификатор фактуре за преузимање (параметар *invoiceId*)

У одговору се, уколико је успешан, у *JSON* формату налази:

- статусни подаци о фактури, где је за излазне фактуре уписано и да ли је фактура успешно издата (пошто обрада издавања фактуре може бити одложена, а идентификатор фактуре смо већ добили)

#### Садржај појединачне издате фактуре

*method:* GET

*path:* /api/publicApi/sales-invoice/{invoiceId}/xml

Минималан скуп параметара је:

- идентификатор фактуре за преузимање (*invoiceId*)

У одговору се, уколико је успешан, у *XML* формату налази:

- омот фактуре који укључује фактуру у UBL формату

#### Статус појединачне примљене фактуре

*method:* GET

*path:* /api/publicApi/purchase-invoice/{invoiceId}

Минималан скуп параметара је:

- идентификатор фактуре за преузимање (*invoiceId*)

У одговору се, уколико је успешан, у *JSON* формату налази:

- статусни подаци о фактури

Садржај појединачне примљене фактуре

*method*: GET

*path*: `/api/publicApi/purchase-invoice/{invoiceId}/xml`

Минималан скуп параметара је:

- идентификатор фактуре за преузимање (*invoiceId*)

У одговору се, уколико је успешан, у XML формату налази:

- омот фактуре који укључује фактуру у UBL формату

Регистрација *callback* операције за нотификације

*method*: POST

*path*: `/api/publicApi/subscribe`

Параметри се преносе у *JSON* речнику у телу *HTTP* захтева (кључеве у речнику сматрамо параметрима)

Овом операцијом се региструје *callback* операција за нотификацију. Регистрација траје до прве ноћне паузе или до поновљене регистрације од клијента са истим идентификатором клијентске стране.

Минималан скуп параметара за операцију регистрације је:

- URL за *callback*, уколико није наведен само ће престати важење претходне регистрације (параметар *url*)

Одговор садржи само информацију да ли је операција успешна.

*Callback* операција користи *POST* метод, са *JSON* речником у телу *HTTP* захтева, где кључеве у речнику сматрамо параметрима.

Минималан скуп параметара у самој *callback* операцији:

- јединствени идентификатор захтева (параметар *requestId*)
- листа догађаја (параметар *eventList*), где сваки елемент листе садржи нову двочлану листу:
  - Тип догађаја (издата фактура, примљена фактура, промена статуса издате фактуре или промена статуса примљене фактуре)
  - идентификатор фактуре

Вредности којима ће се означавати поједини типови догађаја прецизирају се у оквиру крајње техничке спецификације апликативног интерфејса.

Ако *callback* операције врати *HTTP response code 2xx* то значи да су догађаји успешно примљени.

У обради *callback* операције треба само да се обави трансакција евидентирања нотификованих догађаја и да се након успешне те трансакције врати да је нотификација успешно примљена. Обраду евентираних догађаја треба извршавати након тога.

Код позивања и обраде *callback* операције, обе стране треба да правило користе јединствени идентификатор захтева, како је описано у одељку о поузданом преносу података.

Упит промена за издате фактуре

*method:* GET

*path:* /api/publicApi/sales-invoice/changes?date={ date }

Минималан скуп параметара је:

- датум, који не може бити текући дан

У одговору се, уколико је успешан, у *JSON* формату налази:

- листа догађаја, где сваки елемент листе садржи нову двочлану листу:
  - Тип догађаја (издата фактура или промена статуса издате фактуре)
  - идентификатор фактуре

Вредности којима ће се означавати поједини типови догађаја прецизирају се у оквиру крајње техничке спецификације апликативног интерфејса.

Листа догађаја има исту структуру као код *callback* операције што омогућава да се на сличан начин обрађују догађаји добијени упитом и *callback* операцијом.

Упит промена за примљене фактуре

*method:* GET

*path:* /api/publicApi/purchase-invoice/changes?date={ date }

Минималан скуп параметара је:

- датум, који не може бити текући дан

У одговору се, уколико је успешан, у *JSON* формату налази:

- листа догађаја, где сваки елемент листе садржи нову двочлану листу:
  - тип догађаја (примљена фактура или промена статуса примљене фактуре)
  - идентификатор фактуре

Вредности којима ће се означавати поједини типови догађаја прецизирају се у оквиру крајње техничке спецификације апликативног интерфејса.

Листа догађаја има исту структуру као код *callback* операције што омогућава да се на сличан начин обрађују догађаји добијени упитом и *callback* операцијом.

Исказивање потврде или одбијања примљене фактуре

*method:* POST

*path:* /api/publicApi/purchase-invoice/acceptRejectPurchaseInvoice

Параметри се преносе у *JSON* речнику у телу *HTTP* захтева (кључеве у речнику сматрамо параметрима)

Минималан скуп параметара би био:

- јединствени идентификатор захтева (параметар *requestId*)
- идентификатор примљене фактуре (параметар *invoiceId*)
- да ли се прихвата или одбија (параметар *accepted*, *true* или *false*)
- образложење (параметар *comment*)

Јединствени идентификатор захтева треба користити како је описано у одељку о поузданом преносу података.

У одговору само стоји да ли је успешно регистрован исказ.

Исказивања потврде или одбијања примљене фактуре проузрокује позивање *callback* операције за догађај промене статуса примљене фактуре.

Операције за електронско евидентирање обрачуна ПДВ

Операције за електронско евидентирање обрачуна ПДВ се имплементирају по аналогији са операцијама које се односе на издавање фактуре и издате фактуре.