

151026 הנדסת תכנה לאינטרנט – תש"ף

תרגיל 5 להגשה (הגשה בקובץ Word עם zip\rar של פרויקט התרגיל):

(א) יש לקרוא בויקיפדיה על שיטות שונות של שמירת מצב בצד לקוח וצד שרת לפי הרשימה בהרצאות

**(1) מה המטרה של שמירת מצב?**

כדי ליצור חווית משתמש טובה ואיכותית עבור בני האדם אשר רוצים להרגיש שהם מנהלים דו שיח עם האפליקציה בשרת, להרגיש שהיא זוכרת אותם ואת ההעדפות שלהם, נתוני אבטחה וכדומה. שמירת מצב בצד לקוח מטרתה לשמור את המידע שהגיע מהשרת ולהציגו למשתמש באמצעות תוכנה ייעודית, בד"כ הדפדפן. שמירת מצב בצד שרת מטרתה לאחסן ושלמור נתונים של המשתמש לפעם הבאה ולהתחקות אחרי פעולותיו. שיטות לשמירה בצד לקוח- cookies, query String, web storage and view & control state(ASP.net)

שמירה בצד שרת- application state, session state, profile(external DB) cache

**(2) מה היתרונות והחסרונות של שמירת מצב בצד לקוח מול צד שרת?**

יתרונות:

צד שרת- היכולת לספק ממשק מותאם למשתמש, איסוף מאפיינים על הלקוח בכדי לתת לו מענה המותאם אליו אישית, המאפיינים יכולים להיות בקשות ישנות שלו, ההרשאות שלו ומאפיינים אישיים. הקוד הנמצא בצד שרת לא חשוף ללקוח כלל. צד לקוח- במקרה של קריסת שרת ושמירת המצב האחרון, נוכל להציג אותו למרות שהשרת מושבת. אין עיכובים בעיבוד נתונים מקומיים.

חסרונות:

צד שרת- ישנו הצורך לשלוח בקשות נוספות על גבי הרשת כאשר צריך לעדכן נתונים המוצגים למשתמש או להציג מידע חדש דרך הדפדפן, בקשות כאלה מאטות את ממש המשתמש, מעמיסות על עבודת השרת ואף זה יכול לגרום להתנתקות מהשרת ולהפסקת הזרמת המידע ממנו אל המשתמש. צד לקוח- פגם בפן האבטחה, תקורות שונות, עוגיות שונות יכולות להיות לא שמישות על גבי השרתים של המשתמש.

**(3) איך ועל פי אילו סיבות בוחרים שיטות שמירה עבור נתונים שונים בשרת ווב?**

שרת Web איננו שומר מצבים (stateLess), מה שאומר שאחרי יצירת התגובה ושליחתה השרת אינו שומר שום מידע. כל האובייקטים שהשרת הקצה עבור יצירת התגובה נהרסים, כולל כל המידע שהמשתמש שלח לשרת בבקשה וכן המידע שהשרת יצר בתהליך העיבוד ניתן לומר שלאחר העברת התגובה ללקוח, השרת מאבד עניין בלקוח, מנתק את התקשורת עימו והורס את כל האובייקטים והמידע הקשורים בו. אולם, לעיתים היישום מעוניין לשמור נתונים קודמים שהיו בדף לפני שליחתם לשרת. לצורך פתרון בעיה זו נוצרו אובייקטים מיוחדים שכל תפקידם לשמור על המידע בין הבקשות.

**(4) מה זה CORS?**

**CORS** היא דרך מודרנית ובטוחה לאפשר בקשות AJAX בין שני דומיינים. מנגנון מבוסס כותרות HTTP המאפשר לשרת לציין לכל מקורות (דומיין, סכמה או יציאה) מלבד שלו עצמו שממנו דפדפן אמור לאפשר טעינת משאבים. זה בעצם שיתוף משאבים חוצה מקור.

(ב) החלק המעשי- קבצים מצורפים.